St. Cloud State University
theRepository at St. Cloud State

Culminating Projects in Information Assurance | Department of Information Systems

5-2019

Automated Teller Machine Ethernet Traffic Identification to Target Forensics Detection of IP Packets

Brian Volkmuth
bgvolkmuth@stcloudstate.edu

Follow this and additional works at: https://repository.stcloudstate.edu/msia_etds

Recommended Citation
Volkmuth, Brian, "Automated Teller Machine Ethernet Traffic Identification to Target Forensics Detection of IP Packets" (2019). *Culminating Projects in Information Assurance*. 87.
https://repository.stcloudstate.edu/msia_etds/87

This Starred Paper is brought to you for free and open access by the Department of Information Systems at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Information Assurance by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

Automated Teller Machine Ethernet Traffic Identification to Target Forensics Detection of IP Packets

by

Brian Volkmuth

A Starred Paper

Submitted to the Graduate Faculty

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Information Assurance

April, 2019

Starred Paper Committee:
Abdullah Abu Hussein, Chairperson
Lynn A. Collen
Balasubramanian Kasi

Abstract

Over the last few decades, consumers have become accustom to the convenience of Automatic Teller Machines (ATMs) to transfer funds between accounts, provide account balance information and to withdraw cash from savings, checking, and other account types. Along with the convenience and ease of locating an ATM through mobile bank apps, there has been a significant increase in ATM fraud across the globe. Consumer confidence in the ATM, bank and credit card issuer is greatly impacted by the perceived level of security in ATM transactions and the technology behind them. Confronting the risk associated with ATM fraud and limiting its impact is an important issue that face financial institutions as the sophistication of fraud techniques have advanced. Largely the process behind the verification of these transactions has moved from Plain Old Telephone System (POTS) to Ethernet connections to the processors, banks and card issuers. The attack surface has grown, both in size and complexity. These security risks should be prompting the industry to research all attack surfaces, and this research looks specifically the Ethernet packets that make up these types of transactions. In this research, I investigate the packet structure and predictability within ATM Ethernet traffic. Even with the proliferation of retail ATMs in the most common of retail spaces, this attack vector has received little attention.

Acknowledgements

I would like to thank Dr. Abdullah Abu Hussein for his patient help, guidance, and encouragement and Dr. Collen Lynn and Dr. Balasubramanian Kasi for their thoughtful participation on the committee and excellent feedback and of course my wife, Alicia Volkmuth MBA, MA.

Table of Contents

Chapter Page

List of Tables

Table Page

List of Figures

Figure Page

Chapter I: Introduction

Introduction

As with many conveniences of the modern age, the ATM has brought with its popularity the chance of compromised personal data. “Fraud perpetrated at the Automated Teller Machine (ATM) has suddenly and somewhat unexpectedly exploded to the highest rate in two decades” (Sidel, 2015). The uptick in fraud cases is largely due to the new more secure credit and debit cards that banks and credit card companies are rushing to replace the 30-year-old magnetic strip technology. “Criminals ‘know there is still vulnerability [at the ATM] and they are trying to capitalize on it,’ said Owen Wild, director of security marketing at NCR Corp., on the of the largest ATM manufactures” (Sidel, 2015). While the majority of these cases involve capturing the information at the point of reading the card, this research is looking at the possible vulnerability from and on the network that the ATM is connected, the construction of the packets, and the security protocols used in the TCP portion of the packets..

ATMs (or Automated Teller Machines) are what are known as *Alternative Delivery Channels*.

- *Alternative*, because they are alternative to the human teller.
- *Delivery* because they are programmed to deliver a specific set of services.
- *Channel* because in the banking world, all delivery mechanisms need a *channel* via which the delivery would be made (Khan, 2015).

The security of ATM transactions requires a continuous and proactive review of physical components of the machine related to its connection to a network, the security

in the protocols used, and operating system security. The challenges of the market include :

- Constant evolution of threats
- 30-year-old technology still in use in the form of magnetic stripe cards
- Transition to new embedded chip cards
- Prevalent use of legacy operating systems

Analyzing ATM originated Ethernet traffic to create a template for forensics analysis of packets in order to provide a signature for firewalls, determine vulnerabilities based on the characteristics of the traffic, and what follow on work could be recommended, is the goal of this research. This research will use a common off-the-shelf protocol analyzer and determine if a forensics investigator could identify ATM traffic using a packet inspection profile without the knowledge of the sending or receiving IP information. Should a benchmark be possible from this work, there are many other avenues to explore in the retail ATM market.

Problem Statement

We want all of our ATM transactions to be safe and secure even while the transactional data is in transit from the ATM over the internet to the processing servers and back.

Today, more and more ATMs are connecting to their processing servers via the Internet with Ethernet and, in some cases, Wi-Fi connections. There are significant vulnerabilities that need to be explored to create better-protected transactions for consumers. Captured transactional data packets by the use of a packet sniffer could

allow the capture and nefarious use of private information and harm consumers, retailers and the banking industry. We will use packet sniffing software to capture packets from different types of ATM transactions, evaluate them for predictability, and determine the level of ease in finding transactions in a congested production network.

All ATMs connect to a server to process transactions. As seen in Figure 1, the ATM network connects to the cloud as do all of the entities involved in a transaction. The internal bank network is connected to the cloud and pushes account information updates to an ATM Outsource Network Provider. When an ATM or Point of Sale (POS) machine initiates a transaction, the ATM Outsource Network Provider handles the verification of the card and PIN if used, that sufficient funds are available for the requested transaction, among other details. In the past, all ATMs completed transactions via POTS. Today, there are a number of communication technologies being employed including cabled Ethernet, Wireless Ethernet, cellular data, and satellite internet connections. Each of these presents a unique attack surface, but all depend on an Ethernet connection at some point between the ATM and processing server.

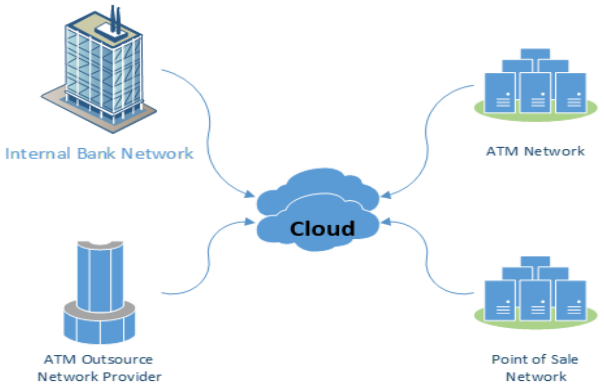


Figure 1. ATM Cloud Transactions

When a transaction is initiated, it follows a flow chart approximated in Figure 2. One of the parts of this study is to determine if it is possible, as the customer works through the decision tree, to see when data is transmitted. In the previous work, data was only transmitted once all the prompts had been satisfied and the customer chose what amount of money was to be withdrawn and from which account. With the advent of EMV technology which included software and hardware upgrade, this may no longer be true.

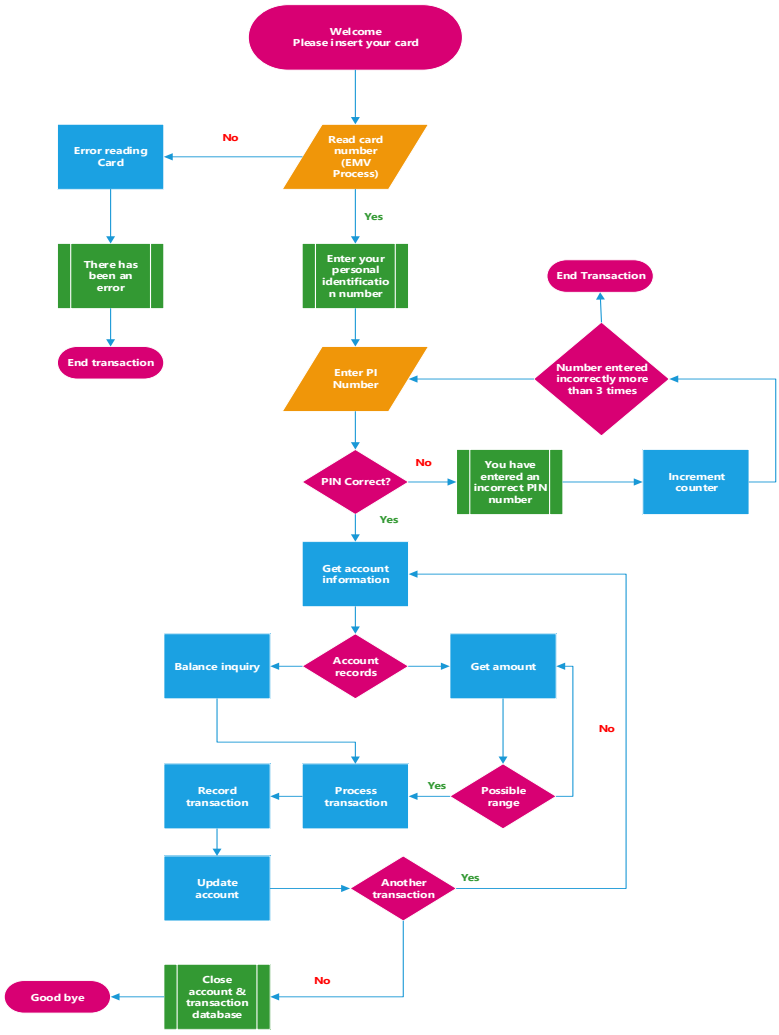


Figure 2. ATM Transaction Flow Chart with Potential Traffic Intercept Points

Nature and Significance of the Problem

The ATM industry has had several government and self-mandated software and equipment updates in a very short period of time. They include changes to the American with Disabilities Act (ADA) elements of an ATM; industry changes to the new and much more secure Euro pay, MasterCard, and Visa (EMV) card technology; and an industry change from dial-up modems to Ethernet and Internet Protocol (IP) based communications. The sheer volume of changes has created a confusing and challenging landscape of regulatory compliance, security gaps, and cost. Combined, this environment has attracted hackers and organized crime groups to pounce on the security gaps and is a world-wide issue. The adaption of current technologies has prompted a change from transmitting transaction data over phone lines to Ethernet connections to the cloud with an encrypted payload.

> Recently, due to heavier computing demands and the falling price of computer-like architectures, ATMs have moved away from custom hardware architectures using microcontrollers and / or application-specific integrated circuits, most of which are based on Intel 8086 architecture, to adopting a hardware architecture that is very similar to a personal computer. Many ATMs are now able to use commercial operating systems such as Microsoft Windows and Linux. Although it is undoubtedly cheaper to use commercial off–the–shelf hardware, it does make ATMs vulnerable to the same sort of problems exhibited by conventional computers. (Adeoye, 2012, p. 457)

Logical attacks on ATMs are on the rise. ATM fraud has increased in sophistication and organization. These organizations have become adept in developing methods and hardware to take advantage of new technologies such as miniaturization, battery life and communications methods like WiFi, Bluetooth, and Near Field Communication (NFC).

> Remote attacks target the ATM networks and attempt to compromise the communication with the host. These attacks are more critical because a hacker does not need to open up the ATMs. As ATM technology knowledge becomes widespread, monitoring systems access through web browsers or TELNET enables an easy access to attackers who can hijack ATM management systems and perform management functions. ATM networks are still vulnerable to similar IP based network attacks. Remote attacks such as Eavesdropping, Spoofing, Denial of Service, Sniffing and Virtual Channel Theft are almost always carried out by criminal organizations. (Chafalon 2012, ¶11-13)

Souvignet, Hatin, Maqua, Tesniere, Léger, and Hormière (2014) reported that there are two main classifications of payment card fraud, card not present and physical fraud. The one we are concerned with here is “physical fraud, where data originates from lost and stolen cards, skimming, shimming, man in the middle attacks, Automated Teller Machine (ATM) reverse engineering, etc.” (p. 143).

Along with the hardware and software changes, there has been a significant shift from dial-up to Ethernet and wireless communication via the internet. This is the focus of this research, specifically looking at packet dumps and identifying ATM specific traffic

based on a multi-point inspection method. It is the norm when doing network forensics or traffic analysis to look at port numbers. In this case the port number is standard, 443, and will aid identification of the target traffic but it will not be the sole metric used. It should be noted that “the traditional approach that relies on transport layer port numbers is largely limited with less than 70% accuracy for application identification” (Alblawi & Jinoh, 2014, p. 4). With that in mind, the decision was made to include a study of the encrypted payload and header information with the exclusion of the destination IP address. To deal with the limitations of port-based inspection, payload information or statistical flow features were used. The payload-based identification inspects application data for traffic classification such as search application-specific signatures. Some interesting work has been done in this area with very good results.

> A machine learning based approach employing simple packet header feature sets and statistical flow feature sets without using the IP addresses, source/destination ports and payload information to unveil encrypted application tunnels in network traffic. We demonstrate the effectiveness of our approach as a forensic analysis tool on two encrypted applications Secure Shell (SSH) and Skype, using traces captured from entirely different networks. (Alshammari & Zincir-Heywood, 2010, p. 1326)

Objective of the Study

The objective of this study is to formulate an initial set of network forensics metrics that determine the predictability of current Automated Teller Machine Ethernet packets transmitted initially on local area networks and eventually over the internet. The

study should determine the number of packets generated during different types of transactions, packet size, packet order and determine what human readable and encrypted components exist in each packet type.

Study Questions/Hypotheses

1. How many packets make up each of the six transactions types?
    a. ATM Startup
    b. Insufficient Funds
    c. Balance Check Incorrect PIN
    d. Balance Check Successful
    e. $20 Withdrawn
    f. $60 Withdrawn
2. Is each packet of predictable length?
3. Is packet length influenced by transaction type?
4. What protocols are used throughout each transaction?
5. What known vulnerabilities exist regarding the packet construction and protocols used in ATM transaction packets?

Limitations of the Study

This study is limited to the developing an understanding of the number, size and repeatable patterns that may occur within several different ATM transaction types. Due to legal limitations, we are unable to do any analysis of the packet payloads and the cryptography contained within them.

Definition of Terms

3DES(“triple DES”). Triple DES applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. In general, TDES was introduced to have three keys. Having a key length of 168 bits: three 56-bit DES keys. When it was discovered that a 56-bit key of DES is not enough to protect from attacks, TDES was chosen as a simple way to enlarge the keyspace without a need to switch to a new algorithm. Triple DES simply extends the key size of DES by applying the algorithm three times in succession with three different keys (Debnath, Agrawal, & Vaishnav 2014).

ATM Vault. The ATM Vault is a secure section of an ATM, separate from the bezel containing the card reader, screen, EPP Pin Pad and receipt printer. It contains the Cash dispensing unit and is secured by either an electronic or dial combination lock.

Americans with Disabilities Act of 1990 (ADA). ADA is a civil rights law that prohibits discrimination against individuals with disabilities in all areas of public life, including jobs, schools, transportation, and all public and private places that are open to the general public. The purpose of the law is to make sure that people with disabilities have the same rights and opportunities as everyone else (ADA National Network).

Automated Teller Machine (ATM). An ATM is a computerized electronic machine that performs basic banking functions (such as handling check deposits or issuing cash withdrawals) (Merriam-Webster's collegiate dictionary 2018).

Bluetooth. A standard for the short-range wireless interconnection of mobile phones, computers, and other electronic devices (Bluetooth, n.d.).

Card Holder. A person who has a credit or debit card.

Card Issuer. A bank or credit union who offers credit cards. When consumers make credit card purchases, the credit card issuer is responsible for sending payments to merchants for purchases made with credit cards from that bank.

Cash Dispensing Unit (CDU). Within the vault of an ATM is the cash dispensing unit. The CDU has electronic sensors that count each bill as it exits the dispenser. It also has sensors that detect damaged bills, doubles or bills stuck together and that detect jammed bills.

Comma Separated Value (CSV). A simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV file format can be imported to and exported from programs that store data in tables such as Microsoft Excel or OpenOffice Calc.

EMV Card Reader. An electronic sensor that reads an EMV chip, magnetic strip, bar code on a credit card, membership card, etc. (ATM of America, n.d.).

EMV Chip (Euro pay, MasterCard, and Visa). A payment instrument technology with an embedded microprocessor chip that store and protect cardholder data (ATM of America, n.d.).

Encrypting PIN Pad (EPPs). A form a component of unattended PIN Entry Devices. Typically, EPPs are used to enter a cardholder's PIN in a secure manner. EPPs are used in conjunction with ATMs, automated fuel dispensers, kiosks, and vending machines. Requirements for these devices are found in security documents published by the Payment Card Industry (ATM of America, n.d.).

Ethernet. The traditional technology for connecting wired local area networks (LANs), enabling devices to communicate with each other via a protocol -- a set of rules or common network language. As a data-link layer protocol in the TCP/IP stack, Ethernet describes how network devices can format and transmit data packets so other devices on the same local or campus area network segment can recognize, receive and process them. An Ethernet cable is the physical, encased wiring over which the data travels (TechTarget Network, n.d.a).

Ethernet Hub. A basic type of wired network device that allows multiple connected computers to communicate via broadcast communication.

Firewall. Part of a computer system or network that is designed to block unauthorized access while permitting outward-bound communication (Firewall, n.d.).

Frame. A digital data transmission unit in computer networking and telecommunication. A frame typically includes synchronization features consisting of a sequence of bits or symbols that indicate to the receiver, the beginning, and end of the payload data within the stream of symbols or bits it receives (Cisco Networking Academy, 2017).

Input/output Board (I/O Board). A circuit board that contains physical connections to peripheral devices. In an ATM these include the receipt printer, card reader, and cash dispensing unit, LCD screen, and EPP Pin Pad (ATM of America, n.d.).

Internet Protocol (IP). The principal communications protocol in the internet protocol suite for relaying datagrams across network boundaries. Its routing function

enables internetworking and essentially establishes the internet (Cisco Networking Academy, 2017).

IP Packet. In computer networks, a pack is a container or box that carries data over a TCP/IP network and internetworks. A packet contains several pieces of information, including the data it is carrying, source and destination IP addresses and other constraints required for quality of service and packet handling (Cisco Networking Academy, 2017).

Local Area Network (LAN). A local area network (LAN) is a group of computers and associated devices that share a common communications line or wireless link to a server. Typically, a LAN encompasses computers and peripherals connected to a server within a distinct geographic area such as an office or a commercial establishment. Computers and other mobile devices use a LAN connection to share resources such as a printer or network storage (TechTarget Network, n.d.b).

Logical Port. In programming, a port is a “logical connection place” and specifically, using the Internet’s protocol, TCP/IP, the way a client program specifies a particular server program on a computer in a network. Higher-level applications that use TCP/IP such as the Web protocol, Hypertext Transfer Protocol, have ports with pre-assigned numbers (Cisco Networking Academy, 2017).

Main Board. Sometimes knows as the motherboard, this is the main printed circuit board found in general purpose microcomputers such as in an ATM and other expandable systems. It holds and allows communication between many of the crucial

electronic components of a system. Such as the central processing unit (CPU) and memory and provides connectors for other peripherals (ATM of America, n.d.).

Near Field Communication (NFC). A technology allowing the short-range wireless intercommunication of mobile phones and other electronic devices for purposes such as making payments.

Network Switch. A computer networking device that connects devices together on a computer network by using packet switching to receive, process, and forward data to the destination device. A network switch is a multi-port network bridge that uses hardware addresses to process and forward data at the data link layer (layer 2) of the OSI model. Some switches can also process data at the network layer (layer 3) by additionally incorporating routing functionality. Such switches are commonly known as layer-3 switches or multilayer switches (Cisco Networking Academy, 2017).

Open Systems Interconnect (OSI) Model. A conceptual model that characterizes and standardized the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology.

- **Layer 1: Physical Layer**. Defines electrical and physical specifications for devices. The physical layer defines the relationship between a device and a transmission medium, such as a copper or optical cable.
- **Layer 2: Data Link Layer**. Provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur in the physical layer.

- **Layer 3: Network Layer**. Provides the functional and procedural means of transferring variable length data sequences from a source host on one network to a destination host on a different network, while maintaining the quality of service requested by the transport layer (in contrast to the data link layer which connects hosts within the same network).
- **Layer 4: Transport Layer**. Provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers of the OSI model. The transport layer controls the reliability of a given link through flow control, segmentation and desegmentation, and error control (Cisco 2017)

Packet Sniffer. A program that can record all network packets that travel past a given network interface, on a given computer, on a network. It can be used to troubleshoot network problems, as well as to extract sensitive information such as credentials from unencrypted login sessions.

Payment Card Industry (PCI). The segment of the financial industry that governs the use of all electronic forms of payment.

Payment Card Industry Security Standards Council (PCI SSC). This oversees policies and technologies behind non-cash payments including transactions involving credit cards, prepaid cards, point-of-sale cards, e-purse, bank debit, and ATM cards.

Payment Processor. A company (often a third party) appointed by a merchant to handle transactions from various channels such as credit cards and debit cards for

merchant acquiring banks. They are usually broken down into two types: front-end and back-end.

Personal Identification Number (PIN). An identifying number allocated to an individual by a bank or other organization and used for validating electronic transactions.

Physical Port. On computer and telecommunication devices, a port is generally a specific place for being physically connected to some other device, usually with a socket and plug (Cisco Networking Academy, 2017).

Plain Old Telephone Service (POTS). A retronym for voice-grade telephone service employing analog signal transmission over copper loops. POTS was the standard service offering from telephone companies from 1876 until 1988. POTS remains the basic form of residential and small business service connection to the telephone network in many parts of the world. The term reflects the technology that has been available since the introduction of the public telephone system in the late 19th century, in a form mostly unchanged despite the introduction of Touch-Tone dialing, electronic telephone exchanges and fiber-optic communication into the public switched telephone network (Plain old telephone service, n.d.).

Router. A networking device that forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet. Data sent through the internet, such as a web page or email, is in the form of data packets. A packet is typically forwarded from one router to another router through the networks that constitute an internetwork until it reaches its destination node. A router is connected to

two or more data lines from different networks. When a data packet comes in on one of the lines, the router reads the network address information in the packet to determine the ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey (Cisco Networking Academy, 2017).

Secure Hash Algorithm 1 (SHA-1). A cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest, typically rendered as a hexadecimal number, 40 digits long.

Secure Hash Algorithm SHA-256. One of a number of cryptographic hash functions. SHA-256 algorithm generates an almost unique, fixed size 256-bit (32-byte) has.

Secure Socket Layer (SSL). A standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. This protocol has now been depreciated by the Internet Engineering Task Force (IETF) and has largely been replaced by Transport Layer Security (Cisco Networking Academy, 2017).

TLSv1.2. The newest SSL protocol version. It introduces new SSL/TLS cipher suites that use the SHA-256 hash algorithm instead of the SHA-1 function, which adds significant strength to the data integrity.

Telnet. A network protocol that allows a user on one computer to log onto another computer that is part of the same or remote network (Cisco Networking Academy, 2017).

Terminal Number. A unique alpha-numeric identifier for each ATM terminal that identifies it to the processing system and is used in combination with the 3DES validation process to make the initial authentication of the terminal to the processing system. It is also used to identify the specific machine during each transaction processed by that terminal (ATM of America, n.d.).

Transport Layer Security (TLS). A protocol that provides privacy and data integrity between two communicating applications (Cisco Networking Academy, 2017).

Uniform Resource Locator (URL). The address of a resource on the Internet. A URL indicates the location of a resource as well as the protocol used to access the resource (Cisco Networking Academy, 2017).

Wi-Fi. The technology for radio wireless local area networking of devices based on the International Electronic and Electrical Engineers (IEEE) 802.11 standards (Cisco Networking Academy, 2017).

Chapter II: Background and Review of Literature

Introduction

There are two areas of importance to this proposal. First, successfully capturing packet data from a variety of ATM transaction types. And second, the determination packet data predictability. In previous work, both aspects were proven to be correct and repeatable. However, this may not be the case with the large number of security and technology changes made in the intervening four years. The specific changes are:

- Updated and patched Windows CE 6.0
- Updated firmware for:
    - EMV Card Reader
    - Cash Dispensing Unit
    - PCI Compliant Key Pad
    - Receipt Printer
- Hardware and software upgrade from a magnetic stripe card reader to EMV chip reader.
    - New card reader
    - New software
    - New internal cables and grounding connections
- Card being swiped to card being dipped and held during the transaction.
- Network settings for FIS changed from a public IP address to a URL.
- Socket layer security changed from SSL to TLSv1.2 or higher.

- FIS server settings changed to reject connection attempts that do not meet the new minimum settings for EMV upgrades.
    - Must use URL
    - TLSv1.2 or higher
    - EMV must be configured on the terminal with appropriate PIDs activated
    - Software version 6.1.19 or higher must be installed.

In the course of this literature review, known ATM attacks and vulnerabilities, as well as the packet sniffing process using common Windows-based open source tools will be explored.

Background Related to the Problem

There has been a significant shift from POTS to Ethernet for secure connections between ATMs and their processing servers. Along with this shift, ATMs have become almost ubiquitous in all retail settings. The manufacturer of the ATM used in this study reports sales in excess of 100,000 units in the United States alone. According to Pymnts.com (2017), the “number of ATMs in the U.S. now stands between 475,000 and 500,000 in 2017” (¶ 1). They are found in convenience stores, bars, restaurants, hotels/motels, and even at short-lived events such as outdoor music festivals and fairs. In some cases, the businesses involved do not own the ATM but do provide the communication infrastructure and in the case of Ethernet, lack the expertise to secure their networks, nor understand what a compromised network looks like. In many locations, access to the network infrastructure is both easy and not monitored. Using a

Packet Sniffer, a compact host device and basic knowledge, the transactions that make up the startup communication between the ATM and processing server and transactions themselves.

Literature Related to the Problem

There is very little to be found in literature related specifically to packet capture of ATM traffic. The vast majority discuss security as it relates to customer satisfaction, physical security, skimmers, malware, and other much more common threats. Therefore, literature has been included literature where security threats from the network are mentioned but rarely described beyond that they are threats found on a network.

The ATM has become as, if not more, ubiquitous than pay telephones were in the past. They can be found in nearly every local eating establishment, convenience store, airport, college, and even hospitals. There are a variety of methods for the ATM to connect to its processing and validating server which include POTS, Ethernet, WiFi, and direct connection to a computer host as a peripheral device.

The benefits of ATM that can be derived from ATM usage are so numerous, some are outlined below:

- Flexible account access allows clients to access their accounts at their convenience.
- MFI (bank) personnel are not required to be present for transactions and have more time to serve clients.
- Increased hours of operation fit client schedules.

- More clients can be reached beyond the branch network, such as in smaller population centers.
- More low-cost funds are available because ATMs make it easier for clients to deposit savings. (Adepju & Alhassan, 2010, p. 4-5)

The security threats are described in how they affect the convenience of ATMs as a service but often not in terms of threats to the user's private data. The convenience that ATMs provide comes with inherent risks both to the consumer as well as the ATM owner.

> ATM fraud is not the sole problem of banks alone; it is a big threat, and it requires coordinated and cooperative action on the part of the bank, customers, and the law enforcement machinery. The ATM frauds not only cause financial loss to banks, but they also undermine customers' confidence in the use of ATMs. (Brunner, Decressin, Hardy, & Kudela, 2004)

The threats to ATMs are both local to the individual machine and can be found in the cloud services that provide authentication and transactional information between the ATM, cardholder and processing agent's infrastructure. "With the introduction of internet technology in recent years, the Internet communication is exposed to unwanted people giving them access to pose different kinds of attacks on ATM Systems" (Twum, Nti, & Assante, 2016, p. 126). Within this statement, the importance of this study can be found and supported. With the significant shift from POTS as the primary means of communication to Ethernet-based technologies, opens up the means to capture transaction data and allow criminals to raid both the ATM, customer's accounts and

consumers' confidence in bank and credit card companies. The gains by criminals have been massive in scope and lighting quick in implementation.

> In 2013, Ghana Commercial Bank (GCB) confirms money theft from an ATM of about GH¢3 million (Obour 2013) and a worldwide gang of criminals stole $45 million in a matter of hours by hacking their way into a database of prepaid debit cards and then draining cash machines around the globe. (Modern Ghana, 2013, p. 127)

The scope of an effect on consumer confidence is an often-studied phenomenon in many countries throughout the world. This provides the motivation to assess the security risks, develop standards and create benchmarks to more efficiently and definitively discover and evaluate compromised aspects of an ATM transaction. "Since it is a day of technology, innovation, and cyber-hacking, so security and privacy of customer's information (Pin code, password, etc.) is a burning issue that has an impact on customer's satisfaction and dissatisfaction" (Hossain, Russel, & Robidas, 2015, p. 68). Customer satisfaction impacts due to security concerns directly relate to this study. While the vast infrastructure comprising the ATM and mobile banking system is largely out of sight to the majority of consumers, the attack surface, even if only looking at the network related portion is large, has many intrusion points and each is likely controlled by an entity with little or no relationship to another nor have the expertise to adequately physically or logically secure their network infrastructure. For many businesses, with regards to ATMs, they relied on POTS communication connections in the past. This

required very little in the way of physical or logical security. Ethernet security is well beyond what they have experienced and often is not a priority.

Literature Related to the Methodology

Capturing and determining packet type, length, protocols used, encryption found if any, and addressing information are the significant data types being sought with this work. To that end, a packet sniffer configured in promiscuous mode will be used.

> IP based sniffing is the most commonly used method of packet sniffing. In this method, a requirement of setting network card into promiscuous mode exist. When network card is set into promiscuous mode, then the host will be able to sniff all packets. (Rupam, Verma, & Singh, 2013, p. 23)

The software Wireshark is a powerful network or protocol analyzer, otherwise known as a packet sniffer. It can be used to look in detail at the structure and content of packets captured from the network interface on the host computer. The features of Wireshark include:

- Data is analyzed either from the wire over the network connection or from data files that have already captured data packets.
- Supports live data reading and analysis for a wide range of networks (including Ethernet, IEEE 802.11, point-to-point Protocol (PPP) and loopback).
- With the help of GUI or other versions, users can browse captured data networks.

- For programmatically editing and converting the captured files to the editcap application, users can use command line switches.
- Display filters are used to filter and organize the data display.
- New protocols can be scrutinized by creating plug-ins. (Wireshark, n.d.)

The end goal of this work is to determine if a benchmark of predictable traffic can be found. In the end, this could be useful in further use of a packet sniffer to monitor network assets.

> The most common goal of packet sniffing is to monitor network assets to detect anomalous behavior and misuse. This concept has been around for nearly twenty years, but only recently has it seen a dramatic rise in popularity and incorporation into the overall information security infrastructure. (Banerjee, Vashishtha, & Saxena, 2010, p. 2)

Due to the dearth of literature on the use of a packet sniffer to specifically capture ATM traffic, the methodology of packet capture is generic in nature at the proposed level of granularity for this study.

Chapter III: Methodology

Introduction

This section outlines creating the test environment used for ATM transactions over Ethernet networks with the use of a packet analyzer. Owing to the limited availability of a real-world e-commerce Ethernet network to perform this experiment, the ATM will be connected to a network in an industrial setting to simulate a congested retail network for this study. We will discuss the setup and design of the network and the structured approaches that will assist in this study.

Design of the Study

To answer the given research question, quantitative methods would be most appropriate to find an answer. For the data samples collected commonly found in the market place retail ATM, a debit card, and running six transaction types:

- Successful Withdrawal $20
- Successful Withdrawal $60
- Successful Balance Check Zero Balance
- Successful Balance Check $110
- Insufficient Funds
- Failed Transaction – Expired Card

Data Collection

The initial set of data will be collected in small network will little overhead or other traffic to obtain an understanding of each of the transmissions more easily. As seen in Figure 3, the ATM is connected to a switch which is then connected to a hub, then to a

router and out to the cloud. The packet sniffing host is connected to the hub to provide access to all communication on the LAN.

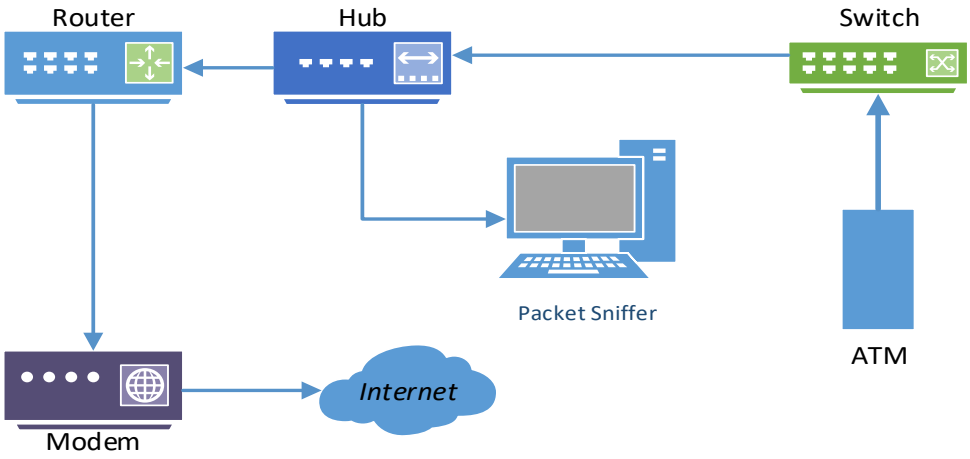


Figure 3. Common ATM Network Diagram

After the first data set has been analyzed and any repeatable metrics have been determined, the second set of data will be collected in a much more congested production network. The ATM will be placed deep within a segmented network to create a condition in which a packet sniffer will have to contend with a much larger number of packets. This could create the possibility of packets with similar characteristics of those found in ATM packets in the first data set.

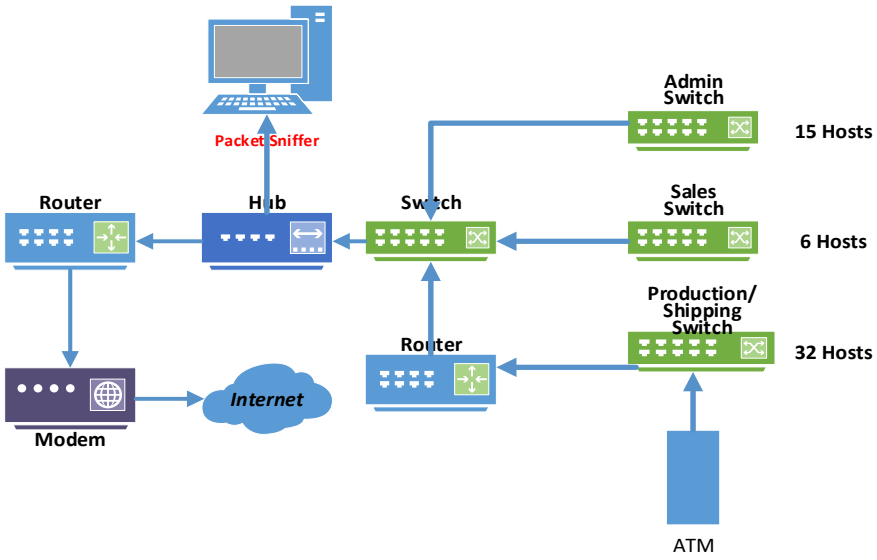


Figure 4. Congested Production Network Packet Capture Topology

Tools and Techniques

To analyze the packet data collected in WireShark, the following information was collected for each transaction and evaluated against each transaction for the following metrics:

- The initial ATM machine start up and connect to the processing server
- Number of packets per transaction
- Size, in bytes, found in each packet

Hardware and Software Environment

This study required the use of a Windows workstation, running Windows 10, Wireshark Network Analyzer v2.6.4, a hub, a border network segment, production network, a retail ATM with the latest OS, software and firmware patches, and a credit card.

Chapter IV: Data Presentation and Analysis

Introduction

The data collected in the research was collected as designed without complications. The process of mining the data for the elements related to this research was also completed without issue. The very capable filters available in WireShark made this process efficient and allowed the results to be saved as a Comma Separated Value (CSV) file, opened in Microsoft Excel and further sorted and analyzed. The results, however, show some expected and unexpected elements that require some manipulating of the data to get to the sought-after information. The process of culling the data is described for each of the research questions as needed.

Data Presentation

Unlike the original research project, the data stream between the ATM and the processing server at startup was captured in order to determine if the new requirements relating to ENV chips, changing to TLSv1.2, and updated and patched OS and PAI software. At the time of the previous work, a more basic authentication process to the server that occurs each time the ATM is powered up, often used POTS versus Ethernet which has since become more prevalent. In addition to a large shift to using Ethernet instead of POTS, the authentication process has grown in complexity and duration. The addition of EMV card readers, TLSv1.2, and the latest PCI compliant keypads are the cause of the new complexity while the OS and PAI software and peripheral firmware versions remain a component of the startup authentication checks but did not add anything new. The new changes created the curiosity of what that looks like from the

perspective of the number, size, and protocols found in the data packets, led to the first data capture. This data set was also used this to verify that the network segment used had a considerable amount of traffic. In Figure 5 below, a truncated display of the first WireShark capture of approximately 2 minutes and 36 seconds during which nearly 73,000 packets were captured.

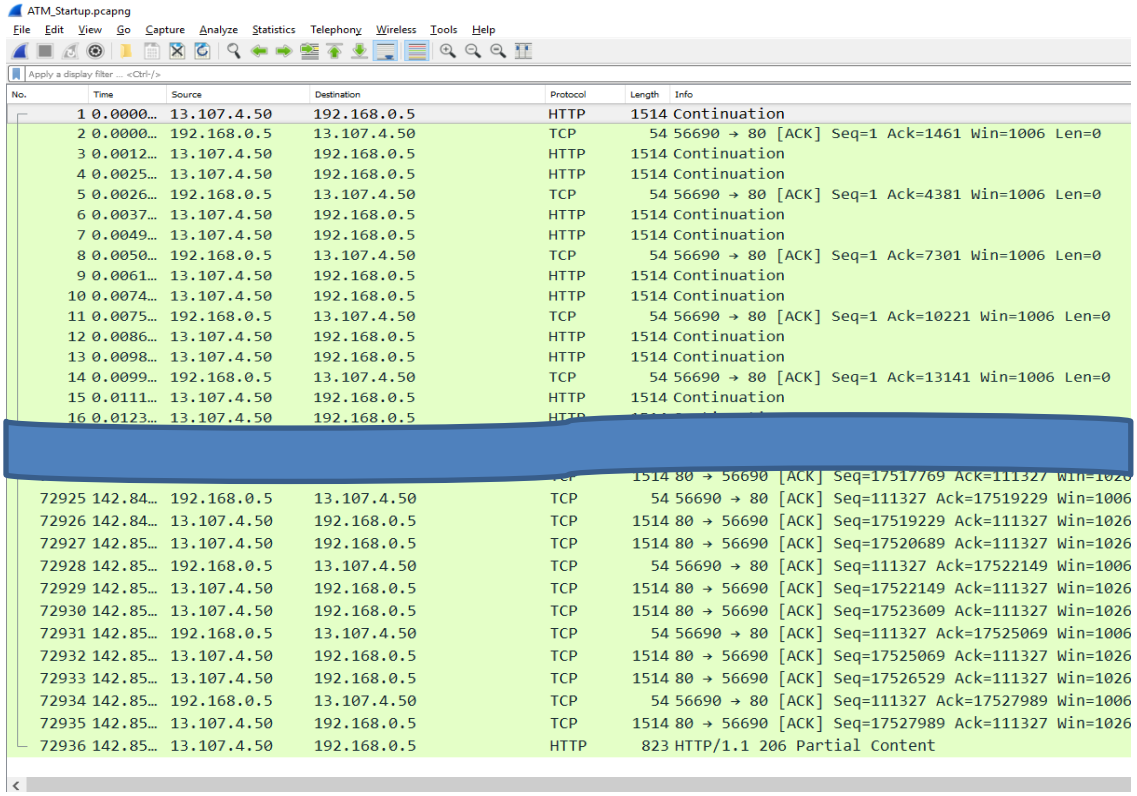


Figure 5. ATM Start-up Authentication Packet Capture

With this initial capture, the authentication to the server can be found, along with the resolved IP address of the server, TCP handshake, and TLSv1.2 Client Hello and other information as seen in Figure 6.

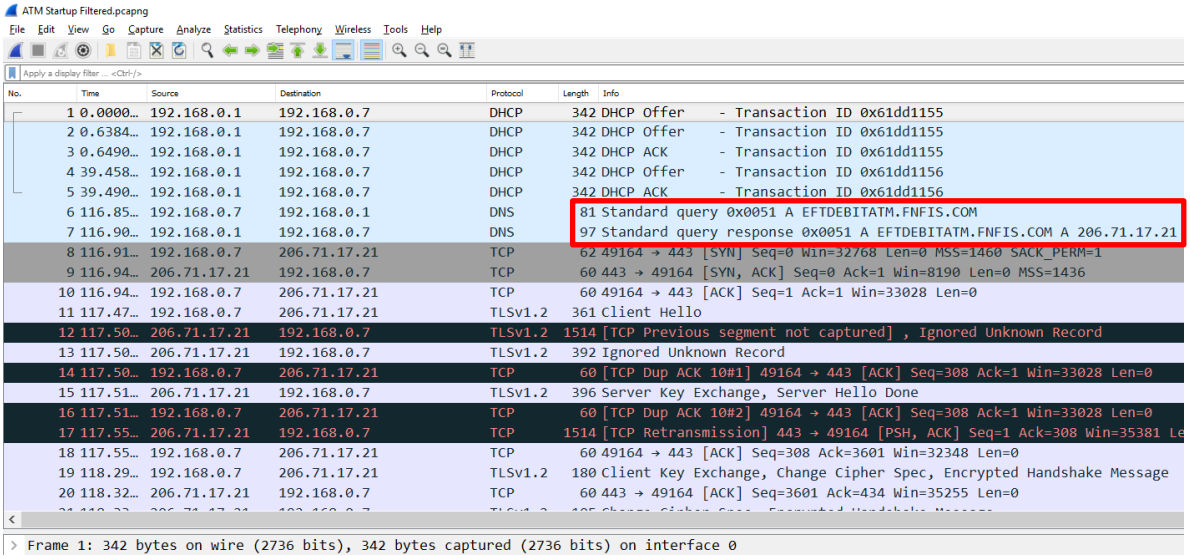


Figure 6. Initial ATM Authentication Packet Capture

This 22-packet data exchange includes updating the processor with the current terminal number, amount of cash in the vault and any other parameters that may have changed. These could include the terminal ID, software version, and firmware versions for any of the peripherals such as the printer, cash dispensing unit (CDU), card reader, or PCI keypad. In the case of the terminal ID or the PCI Keypad, if they had been changed and not re-authenticated by keying in 3DES keys and a secondary authorization to the server via a POTS connection, the machine would not be authenticated and would not enter a transaction ready state and would go out of order. This machine was properly programmed and updated and was allowed to enter a ready state for customers to initiate a transaction.

After the ATM had been verified to be authenticated to the processing server, the test transactions were run. From the initial capture showing the authentication to the server located at IP address of 206.71.17.21, and the ATM had an address of

192.168.0.7. These addresses were then used to filter the packets in each capture to an output that only included packets from each transaction and those addresses. Once the ATM had received DHCP services and dynamically obtained a private IP address, the start-up data was collected showing that IP address. Then the ATM was statically configured with that same private IP address to keep the data collected consistent and efficient. It should be noted that one of the changes from the captures done in 2012 is that the ATM is programmed to connect to a URL requiring a DNS resolution of that URL to an IP address. In the past, ATM's were programmed with the public IP address of the FIS server which removed the step of the DNS resolution. This was one of several new security protocols implemented at the time that EMV cards were required in ATMs. One of those filtered data captures is shown in Figure 7. The filtered captures for each of the six different transactions can be found in the index.

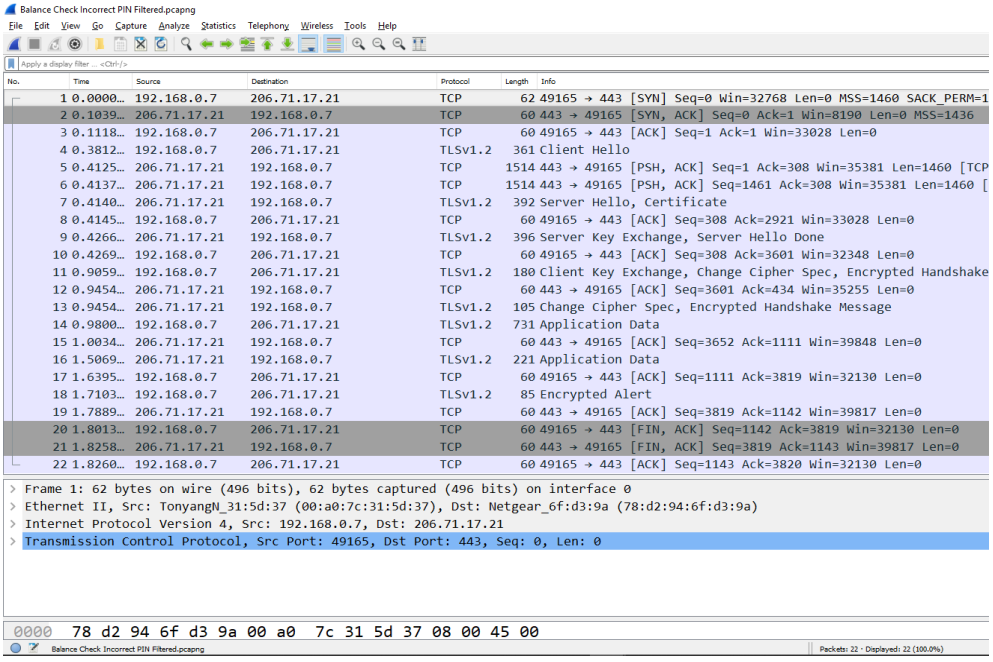


Figure 7. Filtered Packet Capture

For comparison to the previous work which is the basis for the current research questions, Table 1 shows the packet numbers and size for each transaction in the 2012 study. Every transaction type was completely predictable in the number of packets and only had two outliers in packet size out of 85 data points. This confirmed the possibility of creating a network forensics benchmark for ATM Ethernet traffic under the conditions tested and the authenticating server's requirements at that time.

Table 1

Packets Captured for Five Test Transactions

Packet Number	Test 1	Test 2	Test 3	Test 4	Test 5
1	60	62	62	60	60
2	60	60	60	60	60
3	60	60	60	60	60
4	269	269	269	269	269
5	793	793	793	793	793
6	60	60	60	60	60
7	244	244	244	244	244
8	60	60	60	60	60
9	105	105	105	105	105
10	320	320	320	320	320
11	60	60	60	60	60
12	187	187	187	187	187
13	60	60	60	60	60
83	83	83	83	83	83
15	60	60	60	60	60
16	60	60	60	60	60
17	60	60	60	60	60

However, Table 2 created from the current research shows a different result. The number of packets per transaction varies from 21 to 24, and the orderly size of packets found in the original research is not found here either.

Table 2

Packets Captured for Five Test Transactions (2019)

Packet #	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
1	62	62	62	62	62	62
2	60	60	60	60	60	60
3	60	60	60	60	60	60
4	361	361	361	361	361	361
5	1514	1514	1514	1514	1514	1514
6	1514	1514	1514	1514	1514	1514
7	392	392	392	392	392	392
8	60	396	60	60	60	60
9	105	60	396	396	396	396
10	396	180	60	60	60	60
11	60	60	180	180	180	180
12	187	105	60	60	60	60
13	60	731	105	105	105	105
14	180	60	731	60	60	731
15	60	221	60	731	731	60
16	105	60	221	60	60	221
17	731	85	60	221	221	85
18	60	60	85	85	60	60
19	221	60	60	60	85	60
20	60	60	60	60	60	60
21	60	60	60	60	60	60
22	60		60	60	60	
23	60				60	
24	60					

By taking a closer look at the composition of the packets, some assumptions can be made to cull the data a bit further into something more useful for this research. Table 2 shows that the packets with lengths of 60 and 62 can be considered to be overhead

packets. They are acknowledgments to the server for specific frames outlined in red in Figure 8 and can be considered extraneous for this research. The variance in the number of these packets is not a product of the different types of transactions process (i.e., withdrawal, vs. balance check) but varying network infrastructure conditions between the ATM and the processing server that may require the re-transmittal of a packet or a duplicate acknowledgment. This is seen with more clarity when looking at the ATM start-up packet capture which can be seen in Table 6 and is discussed further in this paper. These are indicated by the light blue background in the cells of Table 2.

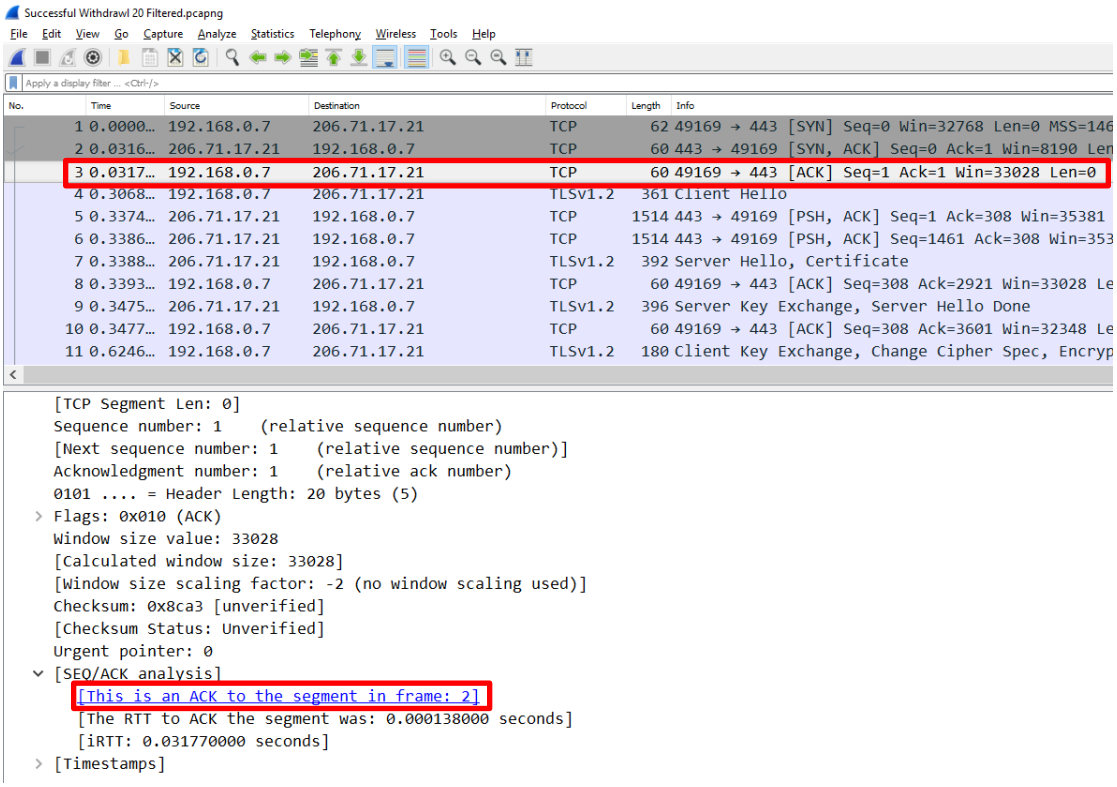


Figure 8. Overhead Packet

When the data is culled to remove the overhead packets, the new table once again shows the predictability of ATM transaction traffic on an Ethernet network. What is

left are the important packets. Table 3 shows the six test transactions and the important packets that make up each transaction.

Table 3

Data Packets with Overhead Packets Removed

Packet #	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
1	62	62	62	62	62	62
2	60	60	60	60	60	60
3	60	60	60	60	60	60
4	361	361	361	361	361	361
5	1514	1514	1514	1514	1514	1514
6	1514	1514	1514	1514	1514	1514
7	392	392	392	392	392	392
8	396	396	396	396	396	396
9	180	180	180	180	180	180
10	105	105	105	105	105	105
11	731	731	731	731	731	731
12	221	221	221	221	221	221
13	85	85	85	85	85	85

Table 4 shows the protocol used, length, and other information about each of the important packets. The packet number and time stamp were removed to allow enough space to display the relevant information. The raw data and truncated tables are available in the index.

Table 4

Typical Transaction Packet Data and Information

Source	Destination	Protocol	Length	Info
192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert

Data Analysis

To answer each of the research questions, each data capture was filtered to only include the packets relevant to the transaction between the ATM and the processing server. For each of the six transactions, the overhead packets were removed leaving only those packets that processed the transaction. Then the packets for each

transaction were compared with regards to their number, size, and order processed. The results can be found in each of the Appendixes. For each of the six transaction types, the appendix includes the Wireshark capture as well as an exported CSV file displayed in Microsoft Excel. Each of the CSV files were then manipulated to remove the overhead packets. Then each of the transaction's packets were compared and this is found specifically in Appendix H.

Chapter V: Results, Conclusion, and Recommendations

Introduction

This research is intended to discover the packet characteristics of Ethernet traffic from a common retail ATM. The results of the research show both some expected results and interesting changes from previous work.

Results

The answer to each of the study questions begins with looking at the general transaction packets. Each transaction, when overhead packets are removed from the analysis, was constructed of 18 packets. The initial SYN-ACK packets, which could be considered overhead, are included in the table below since they are required to set up the communication. Table 5 shows the packets considered for the analysis.

Table 5

Final Packet for Analysis

Source	Destination	Protocol	Length	Info
192.168.0.7	206.71.17.21	TCP	62	49165 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert

For each of the six transaction types in Study Question 1, the packet construction was consistent without deviation of non-overhead packets. Because of the predictability of the packet lengths, no matter the transaction type, it creates the possibility of created a forensics benchmark for ATM Ethernet traffic. The combination of 18 packets with packet lengths of 62, 60, 60, 361, 1514, 1514, 392, 396, 180, 105, 731, 231, and 85

should produce a profile with a high probability of identifying this specific traffic. The results for each research question are as follows:

1. How many packets make up each of the 6 transactions types?
    a. There are 13 packets of consistent length that made up each transaction. Refer to Appendixes A through F.
2. Is each packet of predictable length?
    a. Yes, when the overhead packets were removed, the packets were consistent in length. Refer to Appendix H.
3. Is packet length influenced by transaction type?
    a. No, there is not any variance in packet length noted from one transaction type to another. Refer to Appendix H
4. What protocols were used throughout each transaction?
    a. The TCP protocol was used for each transaction with TLSv1.2 used for encryption and data integrity. Refer to Appendix A through H for the data captures that include the protocols use for each packet and each transaction.
5. What known vulnerabilities exist regarding the packet construction and protocols used?
    a. Packet construction – predictable packet length may be vulnerable to a Teardrop attack, especially with the fragmented packets seen in the packet captures, most notably in the ATM start up capture as seen in Table 6. As noted in an article in Security and Communications Networks,

low-rate DDoS attacks are configured often to use know packet lengths used by a service or service to mask the attack. It was also suggested that this could make the attack more difficult to discover in logs because the traffic has the many of the characteristics of expected packets. “Low-rate Distributed Denial-of-Service (low-rate DDoS) attacks are a new challenge to cyberspace, as the attackers send a large amount of attack packets similar to normal traffic, to throttle legitimate flows. (Zhou, Liao, Yuan, & Zhang, 2017, p. 1).

b. TLSv1.2 – has at least six know vulnerabilities: (Prodromou, 2019).
    i. The Padding Oracle on Downgraded Legacy Encryption (POODLE) attack was published in October 2014 and takes advantage that some servers/clients still support SSL 3.0 for compatibility with legacy systems. There are several named vulnerabilities or offshoots of the POODLE name. Zombie POODLE and GOLDENPOODLE were presented at Black Hat Asia last month.

        > Similar to ROBOT, DROWN and many other vulnerabilities affecting HTTPS, these issues stem from continued use of cryptographic modes which should have been long ago deprecated and yet are inexplicably still supported in TLSv1.2. In this case, the troublesome feature is that TLSv1.2 supports CBC mode ciphersuites (Young, 2019).

ii. The Browser Exploit Against SSL/TLS (BEAST) was disclosed in September 2011 and applies to SSL 3.0 and TLS 1.0 so it affects browsers that support TLS1.0 or earlier protocols. The same mitigation efforts for the POODLE vulnerability apply to BEAST.

iii. The Compression Ration Info-leak Made Easy (CRIME) vulnerability affects TLS compression. The compression method is included in the Client Hello message (see Table 6) and it is optional. It is possible to establish a connection without compression. This appears to be a browser related vulnerability and may not apply to this research. However, it will be mentioned in future work.

iv. The Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext (BREACH) vulnerability is very similar to CRIME but BREACH targets HTTP compression, not TLS compression. This attack is possible even if TLS compression is turned off. As with CRIME, BREACH does not appear to be a vulnerability in this scenario.

v. Heartbleed was a critical vulnerability that was found in the heartbeat extension of the popular OpenSSL library. This extension was used by the previous version of the retail ATM tested and is no longer part of the threat landscape.

For the ATM startup packet, there are some similarities in the number and size of packets. However, the data collected possibly shows some effects of the congested network segment to which the ATM was connected. As seen in Table 6, there are multiple packets that were re-transmitted, and segments not captured shown in cells with type in red. However, it seems that the process of the initial startup communications with the processing server is at least in part, duplicated with each processed transaction.

Table 6

ATM Start-up Capture

No.	Source	Destination	Protocol	Length	Info
1	192.168.0.1	192.168.0.7	DHCP	342	DHCP Offer - Transaction ID 0x61dd1155
2	192.168.0.1	192.168.0.7	DHCP	342	DHCP Offer - Transaction ID 0x61dd1155
3	192.168.0.1	192.168.0.7	DHCP	342	DHCP ACK - Transaction ID 0x61dd1155
4	192.168.0.1	192.168.0.7	DHCP	342	DHCP Offer - Transaction ID 0x61dd1156
5	192.168.0.1	192.168.0.7	DHCP	342	DHCP ACK - Transaction ID 0x61dd1156
6	192.168.0.7	192.168.0.1	DNS	81	Standard query 0x0051 A EFTDEBITATM.FNFIS.COM
7	192.168.0.1	192.168.0.7	DNS	97	Standard query response 0x0051 A EFTDEBITATM.FNFIS.COM A 206.71.17.21
8	192.168.0.7	206.71.17.21	TCP	62	49164 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
9	206.71.17.21	192.168.0.7	TCP	60	443 > 49164 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
10	192.168.0.7	206.71.17.21	TCP	60	49164 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
11	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
12	206.71.17.21	192.168.0.7	TLSv1.2	1514	[TCP Previous segment not captured] , Ignored Unknown Record
13	206.71.17.21	192.168.0.7	TLSv1.2	392	Ignored Unknown Record
14	192.168.0.7	206.71.17.21	TCP	60	[TCP Dup ACK 10#1] 49164 > 443 [ACK] Seq=308 Ack=1 Win=33028 Len=0
15	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
16	192.168.0.7	206.71.17.21	TCP	60	[TCP Dup ACK 10#2] 49164 > 443 [ACK] Seq=308 Ack=1 Win=33028 Len=0
17	206.71.17.21	192.168.0.7	TCP	1514	[TCP Retransmission] 443 > 49164 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460

Table 6 Continued

No.	Source	Destination	Protocol	Length	Info
18	192.168.0.7	206.71.17.21	TCP	60	49164 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
19	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
20	206.71.17.21	192.168.0.7	TCP	60	443 > 49164 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
21	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
22	192.168.0.7	206.71.17.21	TLSv1.2	221	Application Data
23	206.71.17.21	192.168.0.7	TCP	60	443 > 49164 [ACK] Seq=3652 Ack=601 Win=40358 Len=0
24	206.71.17.21	192.168.0.7	TLSv1.2	150	Application Data
25	192.168.0.7	206.71.17.21	TCP	60	49164 > 443 [ACK] Seq=601 Ack=3748 Win=32201 Len=0
26	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
27	192.168.0.7	206.71.17.21	TCP	60	49164 > 443 [FIN, ACK] Seq=632 Ack=3748 Win=32201 Len=0
28	206.71.17.21	192.168.0.7	TCP	60	443 > 49164 [ACK] Seq=3748 Ack=632 Win=40327 Len=0
29	206.71.17.21	192.168.0.7	TCP	60	443 > 49164 [FIN, ACK] Seq=3748 Ack=633 Win=40327 Len=0
30	192.168.0.7	206.71.17.21	TCP	60	49164 > 443 [ACK] Seq=633 Ack=3749 Win=32201 Len=0

Conclusion

The conclusions of this research are surprisingly similar to those of previous work in that the transactions were predictable in packet numbers, size and sequence. In the view of the author, this presents two possibilities. The predictability can be used in forensics efforts to identify ATM traffic should packet inspection be needed to answer questions in an investigation of a compromised ATM. Secondly, it could present an interesting and not commonly known attack surface to be exploited.

Future Work

The results of this research have identified several areas of future work some of which would require gaining permission from the credit card issuing entities who own much of the proprietary infrastructure that retail ATMs connect too.

Beginning with the ATM startup capture, additional packet captures should be run to get a more accurate picture of what packets consistently make up the initial communication with the processing server. It should also be analyzed to determine what similarities if any exists between the startup communications packets and those that make up a customer-initiated transaction.

When choosing the transactions to test, two similar transactions were chosen intentionally with a minor change. For instance, two successful withdrawals but with different dollar amounts were processed. This was done with the idea of future work in mind if permission can be obtained for packet payload inspection and other efforts to identify meaningful data in each packet. In previous work it was noted that packets can contain human-readable data related to the processor.

It would also be useful to conduct cable tapping testing as Ethernet cables often exit the vault portion of the ATM and are easily accessible in a retail setting. Also, it is very common to find that the Ethernet cable servicing the ATM is cut, re-terminated to allow for a switch or hub to be placed near the machine to allow for the connection of other machines such as digital jukeboxes or other internet connected devices, now so ubiquitous in the retail environment.

Lastly, the POODLE and compressions related vulnerabilities should be considered if permission for in-depth packet inspection were obtained. If compression is not used, there seems to be the potential for nefarious actors to, at a minimum, gain actionable intel or at the maximum, compromise a transaction or even be able to redirect traffic to a compromised machine.

References

ADA National Network. (n.d.). *Questions, help, and training on the ADA*. Retrieved from www.adaresources.org/

ATM of America. (n.d.). *Hyosung 1800 manual*. Retrieved from www.atmofamerica.com/Manuals/Hyosung/NH1800.pdf

Adepju, A. S., & Mohammed, E. A. (2010). Challenges of automated teller machine (atm) usage and fraud occurrences in Nigeria: A case study of selected banks in Minna Metropolis. *Journal of Internet Banking and Commerce, 15*(2), 1-10. Retrieved from www.arraydev.com/commerce/jibc/.

Adeoye, O. (2012). Evaluating the performance of two-factor authentication solution in the banking sector. *International Journal of Computer Sciences Issues, 9*(4), 457-462.

Alblawi, U., & Jinoh, K. (2014). A hybrid classifier with a binning method for network application identification. *Journal of Integrated Design and Process Science, 18*(3), 3-22. Retrieved from https://www.researchgate.net/publication/279709222_A_Hybrid_Classifier_with_a_Binning_Method_for_Network_Application_Identification

Alshammari, R., & Zincir-Heywood, A. N. (2010). “Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? *Computer Networks*, *55*(6), 1326-1350. doi:10.1016/j.comnet.2010.12.002

Automated teller machine. (n.d.). In *Merriam Webster*. Retrieved from https://www.merriam-webster.com/dictionary/automated%20teller%20machine

Banerjee, U., Vashishtha, A., & Saxena, M. (2010). Evaluation of the capabilities of wireshark as a tool for intrusion detection. *International Journal of Computer Applications*, *6*(7), 1-5. doi:10.5120/1092-1427.

Bluetooth. (n.d.) In *English Oxford Living Dictionaries*. Retrieved from https://en.oxforddictionaries.com/definition/bluetooth.

Brunner, A., Decressin, J., Hardy, D., & Kudela, B. (2004). *Germany's three-pillar banking system: Cross-country perspectives in Europe*. Washington, DC: International Monetary Fund.

Chafalon, S. (2012). *Bank fraud and ATM security*. Retrieved from http://resources.infosecinstitute.com/bank-fraud-atm-security/

Cisco Networking Academy. (2017). *Routing and switching essentials*. Hoboken, NJ: Cisco Press, 2017.

Debnath, R., Agrawal, P., & Vaishnav, G. (2014). Des, aes and triple des: Symmetric key cryptography algorithm. *International Journal of Science, Engineering and Technology Research*, *3*(3), 652-654. Retrieved from ijsetr.org/wp-content/uploads/2014/03/IJSETR-VOL-3-ISSUE-3-652-654.pdf

Firewall. (n.d.) In *English Oxford Living Dictionaries.* Retrieved from https://en.oxforddictionaries.com/definition/firewall.

Hossain, M. S., Russel, A. H., & Robidas, L. C. (2015). Analysis of factors affecting the customer's satisfaction with reference to ATM services in Dhaka City. *IOSR Journal of Business and Management*, 17(11) 1, 68-75.

Khan, F. (2015). *How does an ATM machine work?* Retrieved from www.quora.com/How-does-an-ATM-machine-work-1

Modern Ghana. (2013). *Hackers steal $45 million in ATM card scam, federal prosecutors say*. Retrieved from www.modernghana.com/news/463043/1/hackers-steal-45-million-in-atm-card-scam-federal.html

Plain old telephone service. (n.d.). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Plain_old_telephone_service.

Prodromou, A. (2019). *TLS security 6: Examples of TLS vulnerabilities and attacks*. Retrieved from https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/143-153

Pymnts.com. (2017). *US ATMs finally see growth*. Retrieved from https://www.pymnts.com/cash/2017/atmia-says-us-atms-finally-see-growth/

Rupam, A., Verma, A., & Singh, A. (2013). An approach to detect packets using packet sniffing. *International Journal of Computer Science and Engineering Survey*, *4*(3), 21-33. doi:10.5121/ijcses.2013.4302.

Sidel, R. (2015). *Theft of debit-card data from ATMs soars*. Retrieved from www.wsj.com/articles/theft-of-debit-card-data-from-atms-soars-1432078912

Souvignet, T., Hatin, J., Maqua, F., Tesniere, D., Léger, P., & Hormière, R. (2014). Payment card forensic analysis: From concepts to desktop and mobile analysis tools. *Digital Investigation, 11*(3), 143-153.

TechTarget Network. (n.d.a). *What is ethernet?* Retrieved from https://searchnetworking.techtarget.com/definition/Ethernet.

TechTarget Network. (n.d.b). *What is local area network (LAN)*? Retrieved from searchnetworking.techtarget.com/definition/local-area-network-LAN

Twum, F., Nti, I. K., & Asante, M. (2016). Improving security levels in automatic teller machines (ATM) using multifactor authentication. *International Journal of Science and Engineering Applications*, *5*(3), 126-134. doi:10.7753/ijsea0503.1003.

Wireshark. (n.d.). In *Techopedia*. Retrieved from www.techopedia.com/definition/25325/wireshark

Young, C. (2019). *Introducing Zombie POODLE and GOLDENDOODLE*. Retrieved from https://www.tripwire.com/state-of-security/vulnerability-management/zombie-poodle-goldendoodle/

Zhou, L., Liao, M., Yuan, C., & Zhang, H. (2017). Low-rate DDoS attack detection using expectation of packet size. *Security and Communication Networks, 2017*(1), 1-14. doi:10.1155/2017/3691629

Appendices

Appendix A
Successful Withdrawal $20

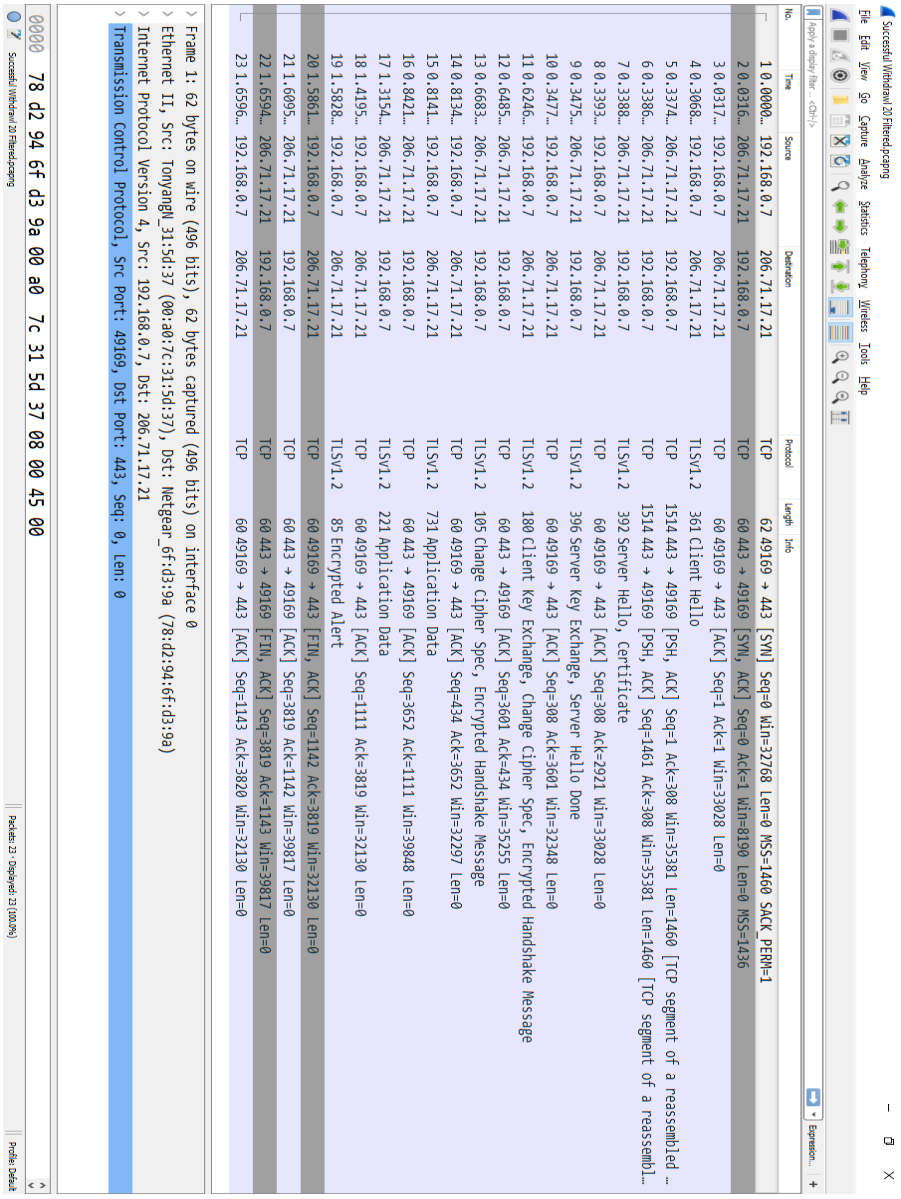


Figure 9. Successful Withdrawal $20 (Wireshark)

	A	B	C	D	E	F	G
3	2	0.031632	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
4	3	0.03177	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
5	4	0.30683	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
6	5	0.337405	206.71.17.21	192.168.0.7	TCP	1514	443 > 49169 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	6	0.338608	206.71.17.21	192.168.0.7	TCP	1514	443 > 49169 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
8	7	0.338801	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
9	8	0.339371	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
10	9	0.347535	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
11	10	0.347736	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
12	11	0.62462	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
13	12	0.648559	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
14	13	0.668324	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
15	14	0.813417	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=434 Ack=3652 Win=32297 Len=0
16	15	0.814157	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
17	16	0.842189	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
18	17	1.31549	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
19	18	1.41955	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
20	19	1.582867	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
21	20	1.586143	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
22	21	1.609597	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
23	22	1.659424	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
24	23	1.659695	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0
25							
26							
27							
28							
29							

Figure 10. Successful Withdrawal $20 (CSV)

Appendix B
Successful Withdrawal $60

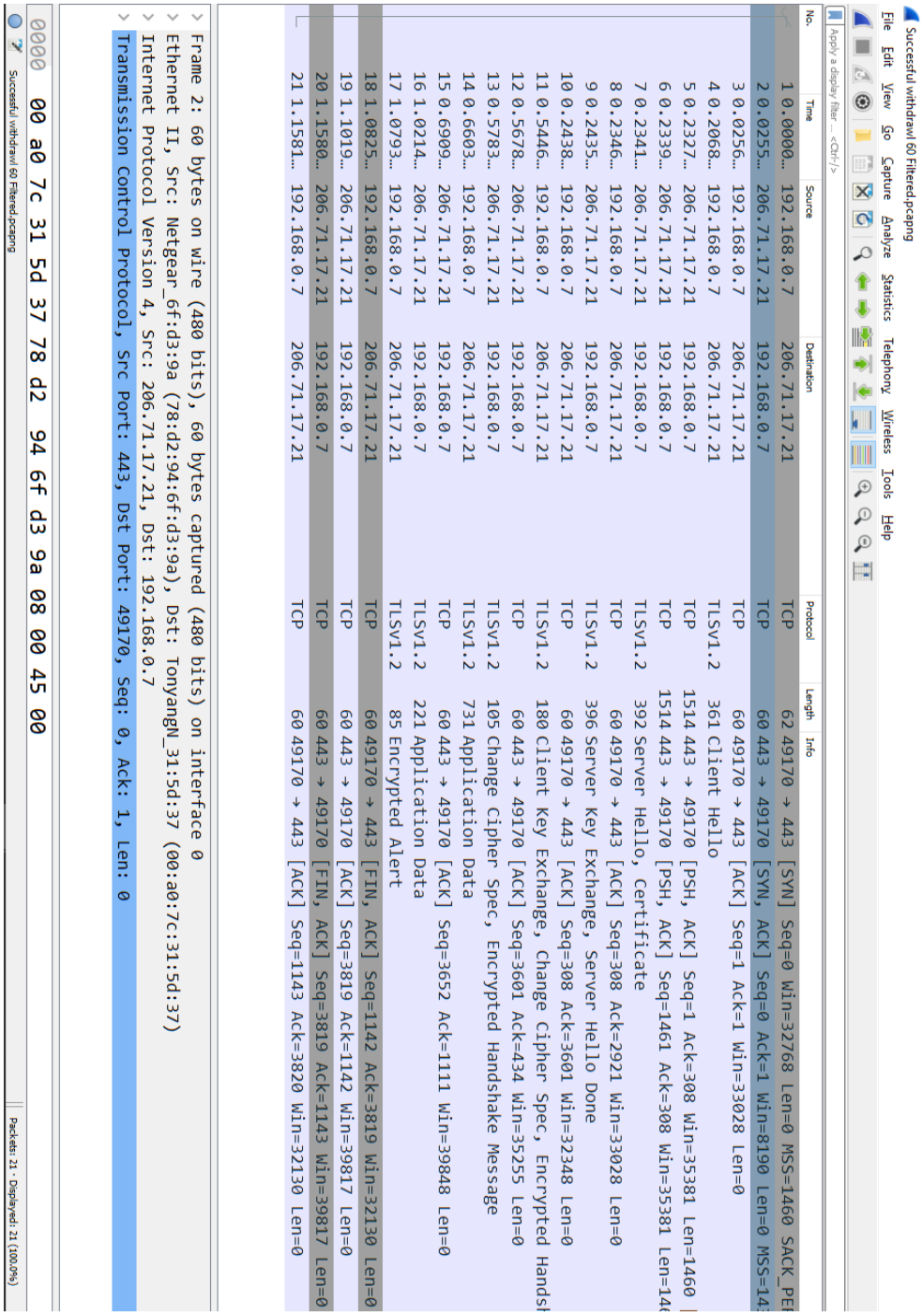
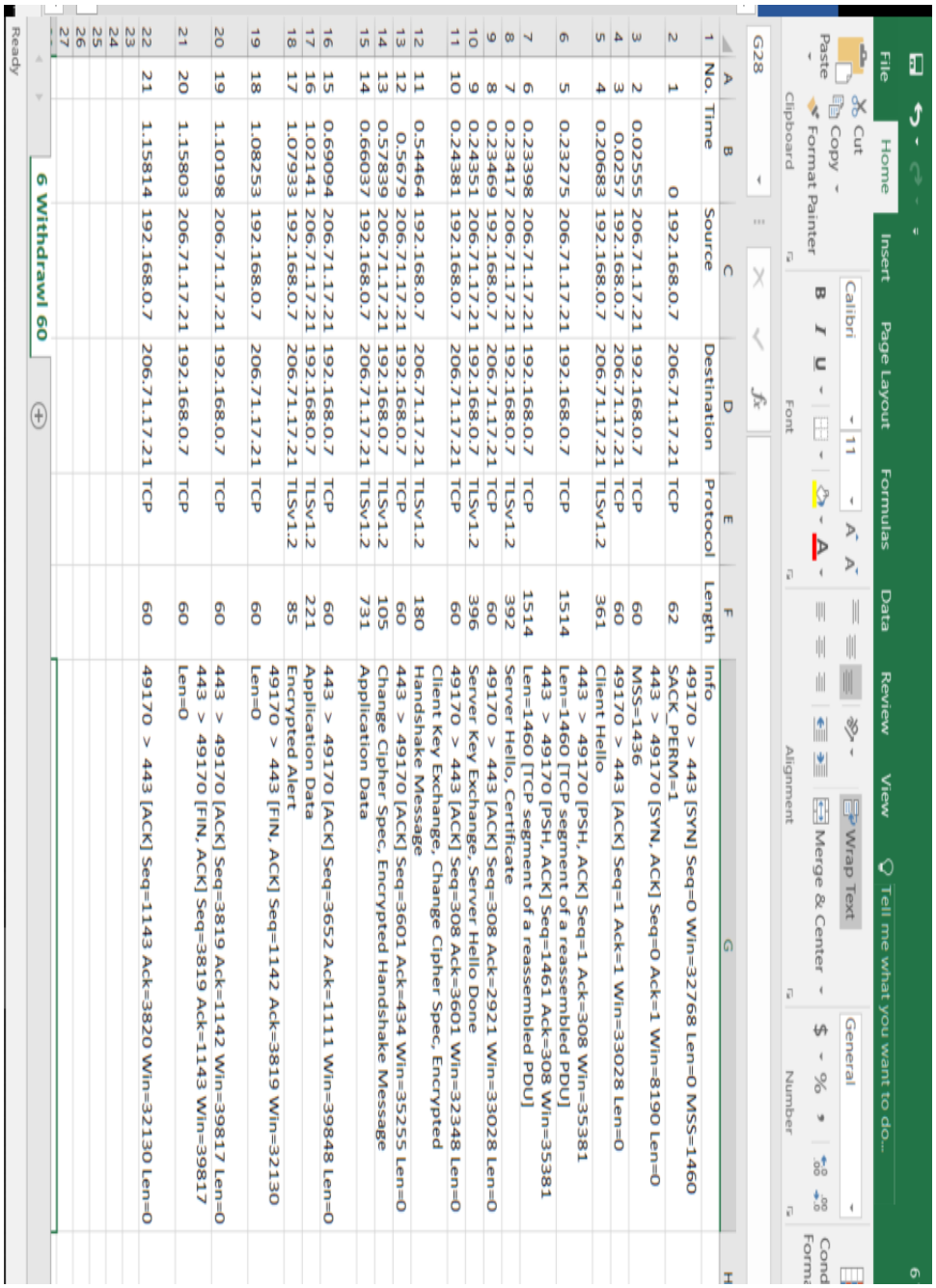


Figure 11. Successful Withdrawal of $60 (Wireshark)



No.	Time	Source	Destination	Protocol	Length	Info
1	0	192.168.0.7	206.71.17.21	TCP	62	49170 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
2	0.02555	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
3	0.0257	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
4	0.20683	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
5	0.23275	206.71.17.21	192.168.0.7	TCP	1514	443 > 49170 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	0.23398	206.71.17.21	192.168.0.7	TCP	1514	443 > 49170 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	0.23417	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
8	0.23469	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	0.24351	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
10	0.24381	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
11	0.54464	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.5679	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
13	0.57839	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
14	0.66037	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	0.69094	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	1.02141	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	1.07933	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
18	1.08253	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
19	1.10198	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
20	1.15803	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
21	1.15814	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0

Figure 12. Successful Withdrawal of $60 (CSV)

Appendix C
Balance Check Successful Zero Balance

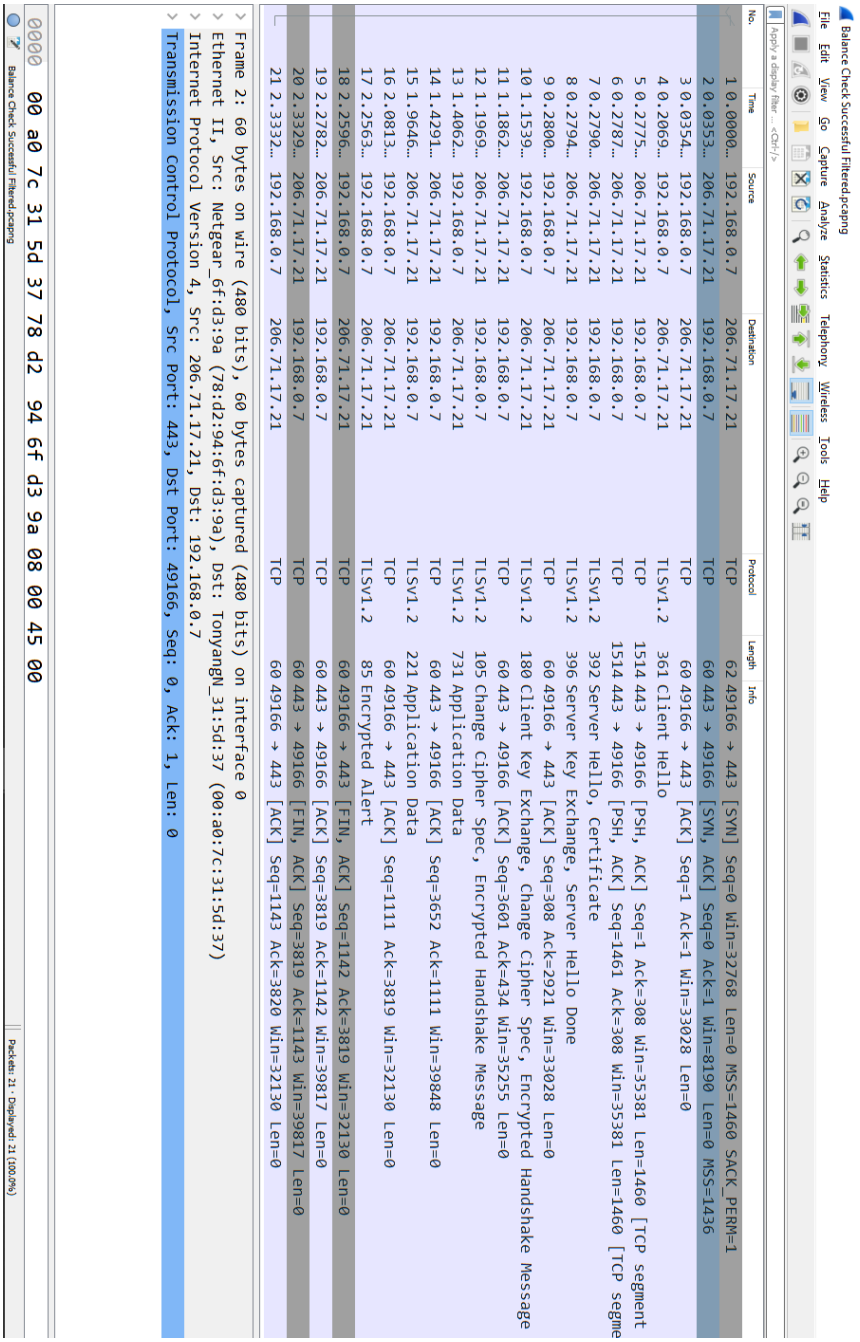


Figure 13. Balance Check Successful Zero Balance (Wireshark)

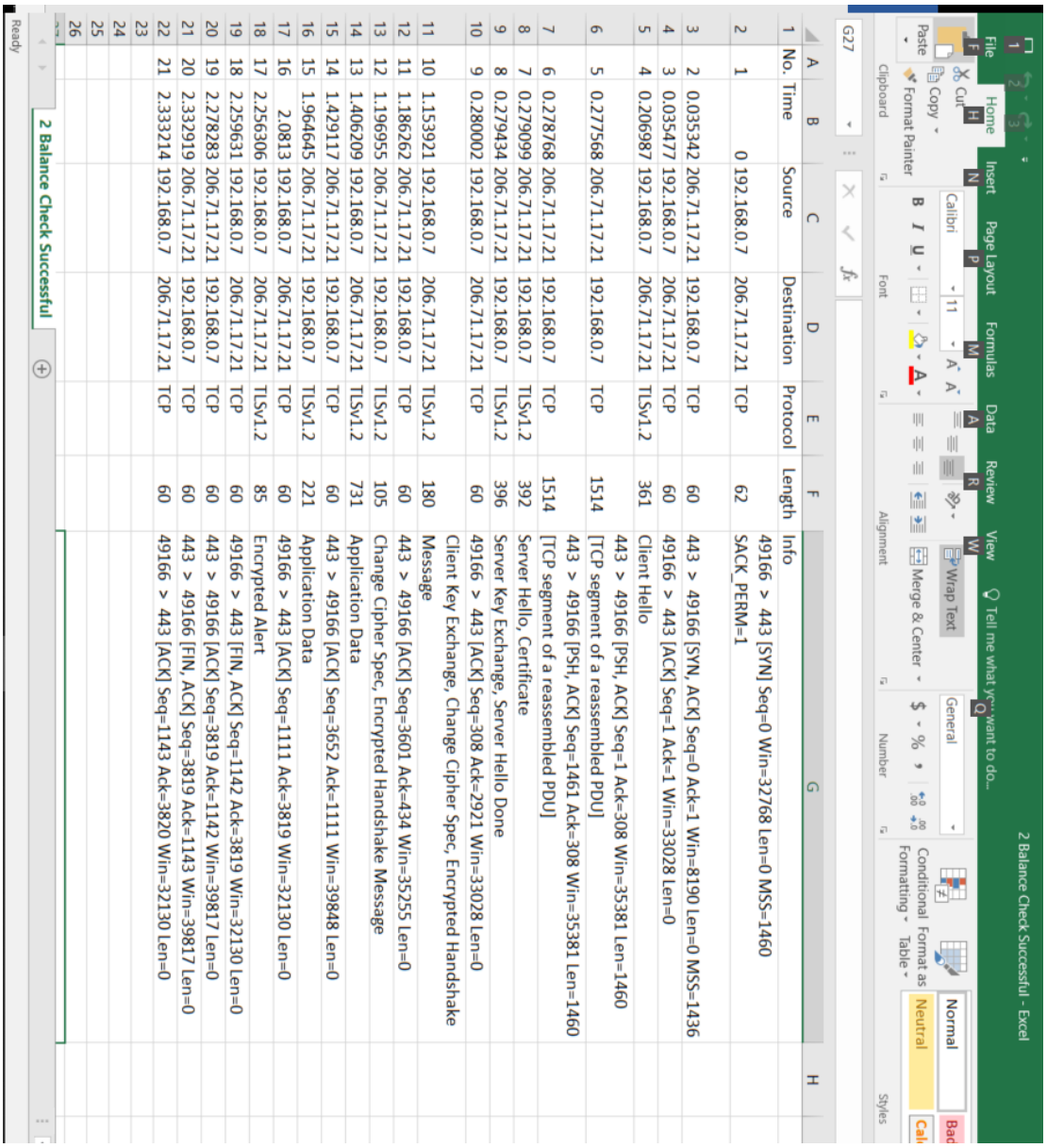


Figure 14. Balance Check Successful Zero Balance (CSV)

Appendix D
Insufficient Funds

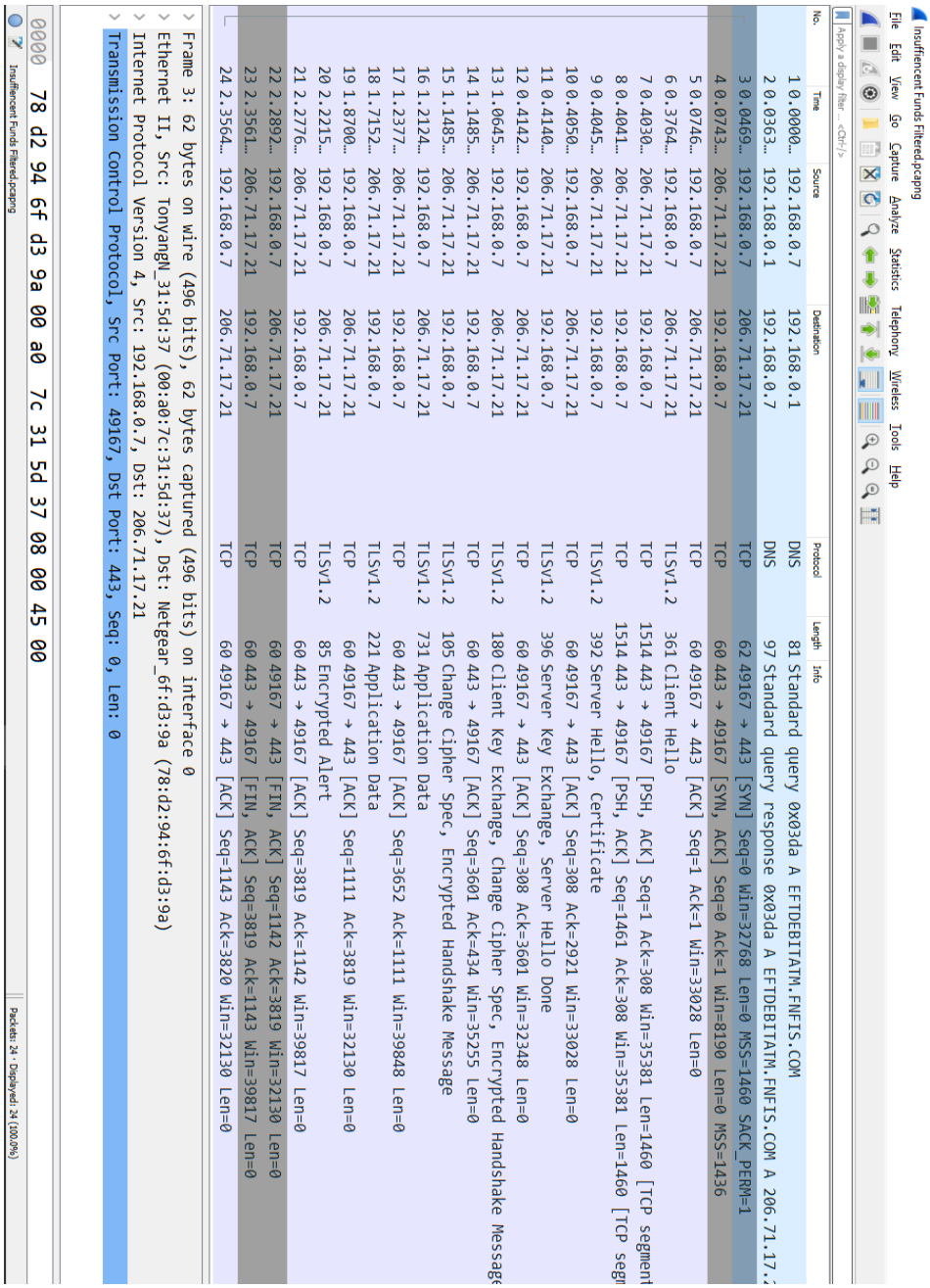


Figure 15. Insufficient Funds (Wireshark)

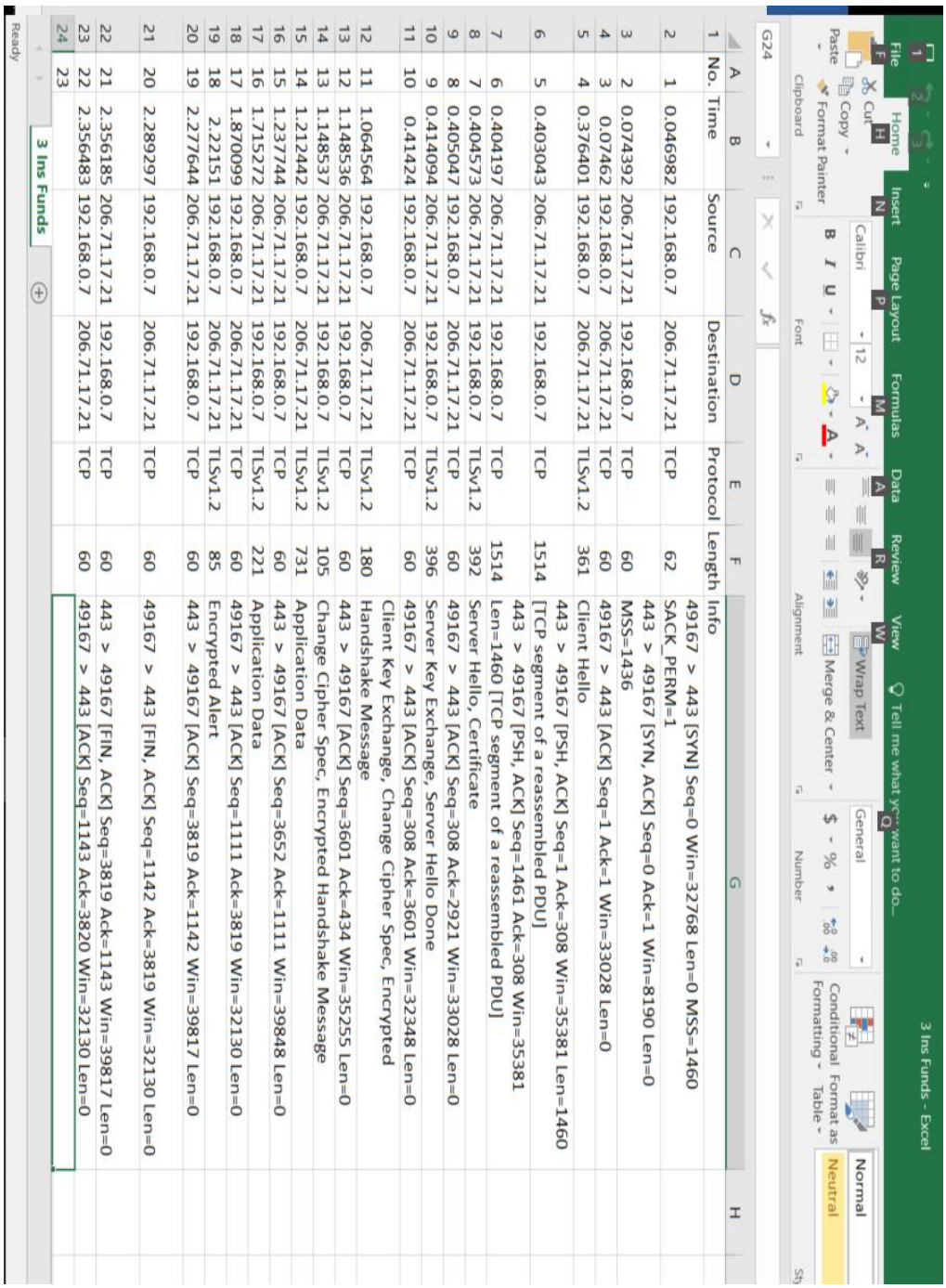


Figure 16. Insufficient Funds (CSV)

Appendix E
Balance Check, Wrong PIN

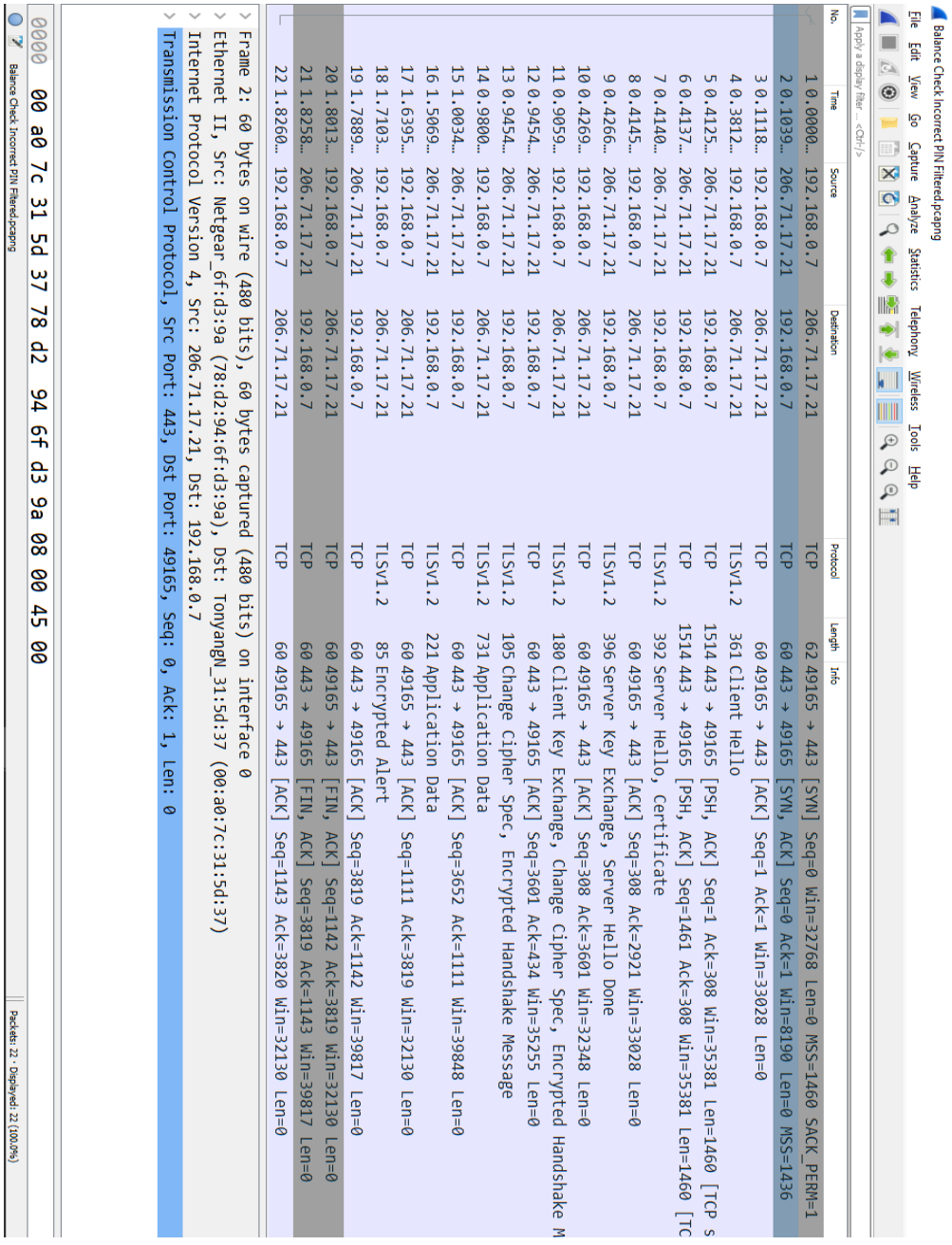


Figure 17. Balance Check, Wrong Pin (Wireshark)

1 Balance Check Incorrect PIN - Excel

No.	Time	Source	Destination	Protocol	Length	Info
1	0	192.168.0.7	206.71.17.21	TCP	62	49165 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
2	0.103904	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
3	0.111887	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
4	0.381249	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
5	0.41254	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	0.413755	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	0.414071	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
8	0.414577	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	0.42667	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
10	0.426977	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
11	0.905932	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.945464	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
13	0.945491	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
14	0.98003	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	1.003492	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	1.506922	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	1.639561	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
18	1.710325	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
19	1.788997	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
20	1.801319	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
21	1.825801	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
22	1.826029	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0

Figure 18. Balance Check, Wrong Pin (CSV)

Appendix F
Successful Balance Check $110

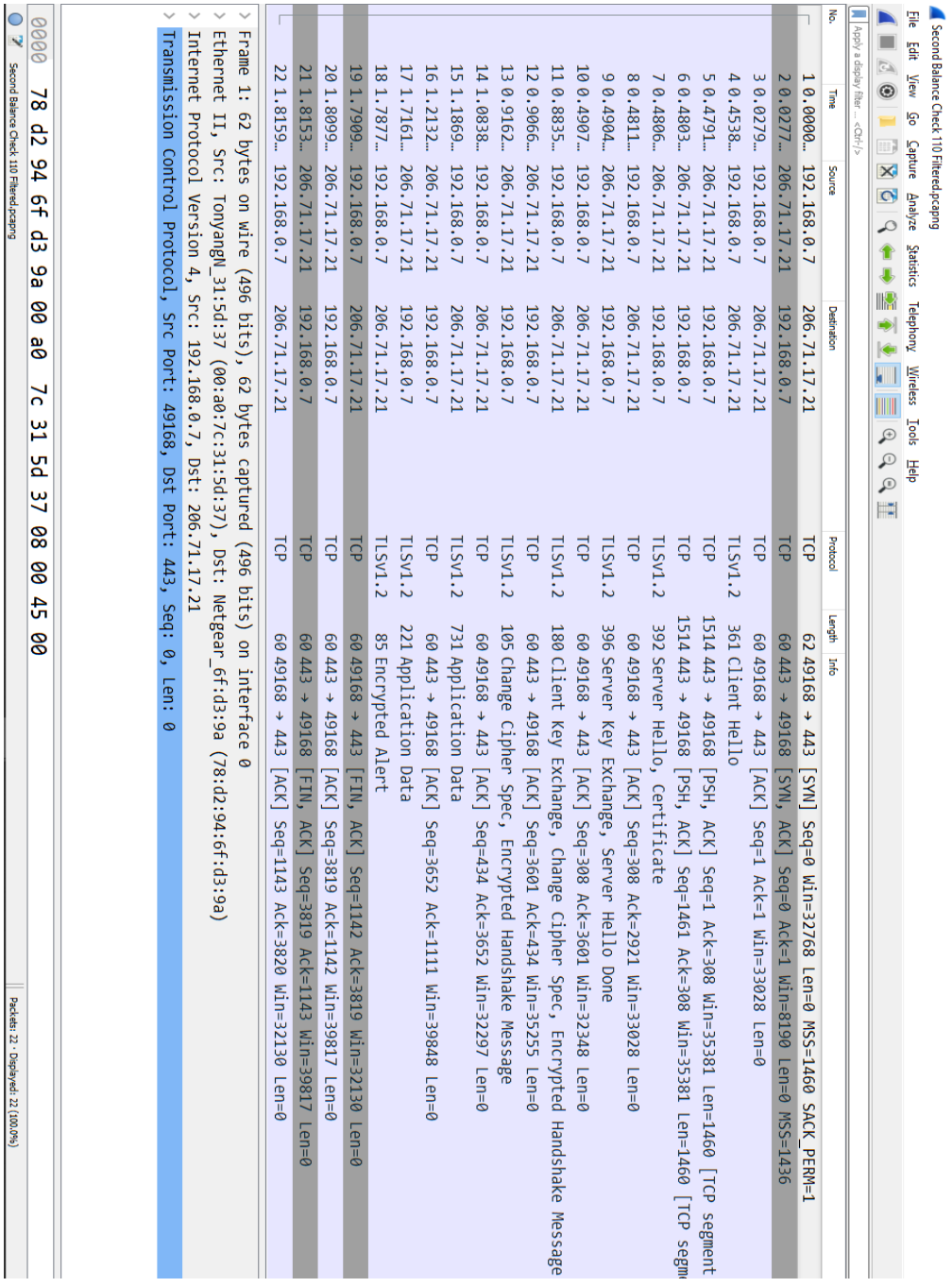
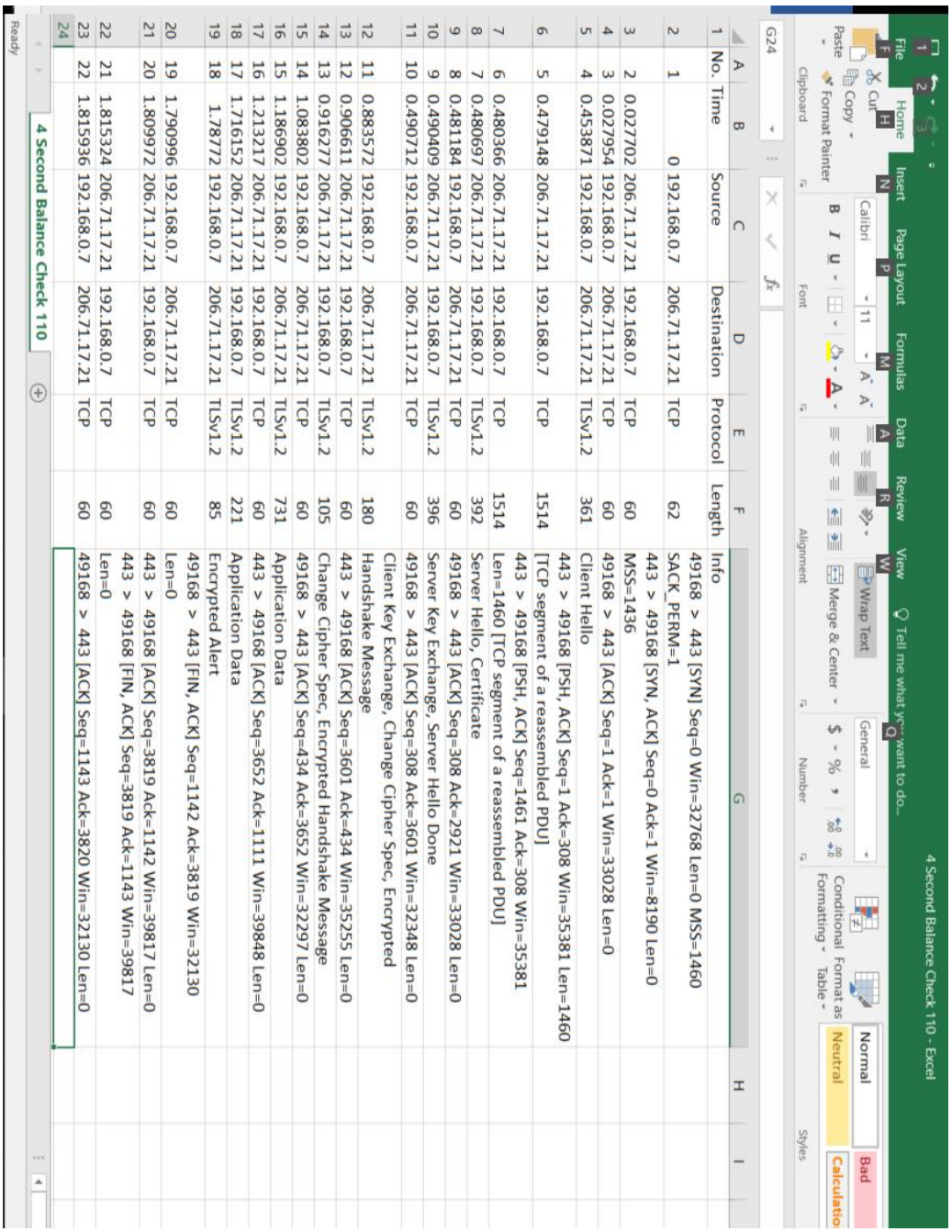


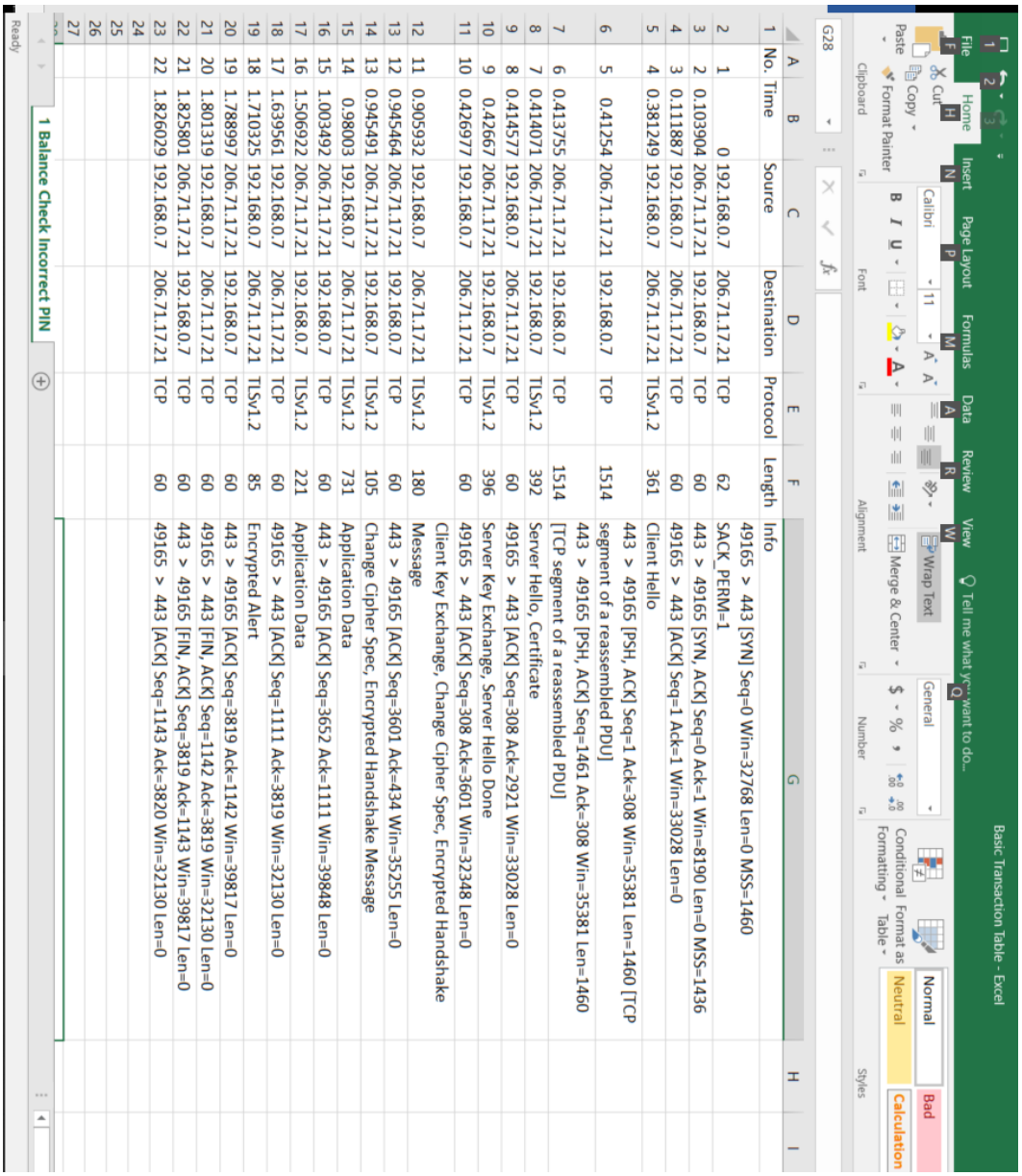
Figure 19. Successful Balance Check $110 (Wireshark)



No.	Time	Source	Destination	Protocol	Length	Info
1	0	192.168.0.7	206.71.17.21	TCP	62	49168 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
2	0.027702	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
3	0.027954	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
4	0.453871	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
5	0.479148	206.71.17.21	192.168.0.7	TCP	1514	443 > 49168 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	0.480366	206.71.17.21	192.168.0.7	TCP	1514	443 > 49168 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	0.480697	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
8	0.481184	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	0.490409	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
10	0.490712	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
11	0.883572	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.906611	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
13	0.916277	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
14	1.083802	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=434 Ack=3652 Win=32297 Len=0
15	1.186902	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
16	1.213217	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
17	1.716152	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
18	1.78772	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
19	1.790996	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
20	1.809972	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
21	1.815324	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
22	1.815936	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0

Figure 20. Successful Balance Check $110 (CSV)

Appendix G
Basic Transaction Data



No.	Time	Source	Destination	Protocol	Length	Info
1	0	192.168.0.7	206.71.17.21	TCP	62	49165 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
2	0.103904	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
3	0.111887	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
4	0.381249	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
5	0.41254	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	0.413755	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	0.414071	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
8	0.414577	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	0.42667	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
10	0.426977	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
11	0.905932	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.945464	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
13	0.945491	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
14	0.98003	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	1.003492	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	1.506922	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	1.639561	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
18	1.710325	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
19	1.788997	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
20	1.801319	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
21	1.825801	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
22	1.826029	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0

Figure 21. Basic Transaction Data (CSV)

Appendix H
Packet Sequence Comparison by Packet Number

Table 7

Packet Sequence Comparison by Packet Number

No.	Source	Destination	Protocol	Length	Info
1	192.168.0.7	206.71.17.21	TCP	62	49165 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
1	192.168.0.7	206.71.17.21	TCP	62	49166 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
1	192.168.0.7	206.71.17.21	TCP	62	49167 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
1	192.168.0.7	206.71.17.21	TCP	62	49168 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
1	192.168.0.7	206.71.17.21	TCP	62	49169 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
1	192.168.0.7	206.71.17.21	TCP	62	49170 > 443 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 SACK_PERM=1
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49166 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49167 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
2	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1436
3	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
3	192.168.0.7	206.71.17.21	TCP	60	49166 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0

3	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
3	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
3	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
3	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=1 Ack=1 Win=33028 Len=0
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
4	192.168.0.7	206.71.17.21	TLSv1.2	361	Client Hello
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49166 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49167 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49168 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49169 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
5	206.71.17.21	192.168.0.7	TCP	1514	443 > 49170 [PSH, ACK] Seq=1 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49165 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]

6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49166 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49167 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49168 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49169 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
6	206.71.17.21	192.168.0.7	TCP	1514	443 > 49170 [PSH, ACK] Seq=1461 Ack=308 Win=35381 Len=1460 [TCP segment of a reassembled PDU]
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
7	206.71.17.21	192.168.0.7	TLSv1.2	392	Server Hello, Certificate
8	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
8	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
8	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
8	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
8	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
8	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done

9	192.168.0.7	206.71.17.21	TCP	60	49166 > 443 [ACK] Seq=308 Ack=2921 Win=33028 Len=0
9	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
9	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
9	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
9	206.71.17.21	192.168.0.7	TLSv1.2	396	Server Key Exchange, Server Hello Done
10	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
10	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
10	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
10	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
10	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=308 Ack=3601 Win=32348 Len=0
11	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
11	206.71.17.21	192.168.0.7	TCP	60	443 > 49166 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
11	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
11	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
11	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
11	192.168.0.7	206.71.17.21	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3601 Ack=434 Win=35255 Len=0

12	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
12	206.71.17.21	192.168.0.7	TCP	60	443 > 49167 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
12	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
12	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
12	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3601 Ack=434 Win=35255 Len=0
13	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
13	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
13	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
13	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
13	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
13	206.71.17.21	192.168.0.7	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
14	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
14	206.71.17.21	192.168.0.7	TCP	60	443 > 49166 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
14	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
14	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=434 Ack=3652 Win=32297 Len=0
14	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=434 Ack=3652 Win=32297 Len=0
14	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
15	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
15	206.71.17.21	192.168.0.7	TCP	60	443 > 49167 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0

15	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	192.168.0.7	206.71.17.21	TLSv1.2	731	Application Data
15	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
16	192.168.0.7	206.71.17.21	TCP	60	49166 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
16	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
16	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3652 Ack=1111 Win=39848 Len=0
16	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
17	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
17	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
17	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	206.71.17.21	192.168.0.7	TLSv1.2	221	Application Data
17	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
18	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
18	192.168.0.7	206.71.17.21	TCP	60	49166 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
18	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
18	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
18	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1111 Ack=3819 Win=32130 Len=0
18	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
19	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
19	206.71.17.21	192.168.0.7	TCP	60	443 > 49166 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0

19	206.71.17.21	192.168.0.7	TCP	60	443 > 49167 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
19	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
19	192.168.0.7	206.71.17.21	TLSv1.2	85	Encrypted Alert
19	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
20	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
20	206.71.17.21	192.168.0.7	TCP	60	443 > 49166 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
20	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
20	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
20	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [FIN, ACK] Seq=1142 Ack=3819 Win=32130 Len=0
20	206.71.17.21	192.168.0.7	TCP	60	443 > 49170 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
21	206.71.17.21	192.168.0.7	TCP	60	443 > 49165 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
21	192.168.0.7	206.71.17.21	TCP	60	49166 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0
21	206.71.17.21	192.168.0.7	TCP	60	443 > 49167 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
21	206.71.17.21	192.168.0.7	TCP	60	443 > 49168 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
21	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [ACK] Seq=3819 Ack=1142 Win=39817 Len=0
21	192.168.0.7	206.71.17.21	TCP	60	49170 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0
22	192.168.0.7	206.71.17.21	TCP	60	49165 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0
22	192.168.0.7	206.71.17.21	TCP	60	49167 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0

22	192.168.0.7	206.71.17.21	TCP	60	49168 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0
22	206.71.17.21	192.168.0.7	TCP	60	443 > 49169 [FIN, ACK] Seq=3819 Ack=1143 Win=39817 Len=0
23	192.168.0.7	206.71.17.21	TCP	60	49169 > 443 [ACK] Seq=1143 Ack=3820 Win=32130 Len=0